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.
<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.</td>
<td>November 27, 2017</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jeanne Altmann, PhD.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Susan C. Alberts, PhD.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Leah Gerber</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ER Diagram layout and conversion to Dia</td>
<td>Leah Gerber</td>
<td>November 27, 2017</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Anne Hubbard</td>
<td>November 27, 2017</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Karl Pinc</td>
<td>November 27, 2017</td>
<td></td>
</tr>
</tbody>
</table>
Contents

1 Babase Summarized 1

2 The Babase ER Diagrams 1

3 The Babase Views 22
   3.1 The ACTOR_ACTEES View 22
   3.2 The ANESTH_STATS View 23
   3.3 The BODYTEMP_STATS View 24
   3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views 25
   3.5 The CHEST_STATS View 26
   3.6 The CROWN_RUMP_STATS View 27
   3.7 The CYCLES_SEXSkins and CYCLES_SEXSkins_SORTED Views 27
   3.8 The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views 28
   3.9 The DSAMPLES View 30
   3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views 31
   3.11 The DENT_CODES View 33
   3.12 The DENT_SITES View 36
   3.13 The INTERACT and INTERACT_SORTED Views 38
   3.14 The MATERNITIES View 39
   3.15 The MIN_MAXS View 41
   3.16 The MIN_MAXS_SORTED View 43
   3.17 The MPI_EVENTS View 46
   3.18 The MTD_CYCLES View 48
   3.19 The PARENTS View 50
   3.20 The PCSKINS_SORTED View 51
   3.21 The PCV_STATS View 52
   3.22 The POINTS and POINTS_SORTED Views 53
   3.23 The POTENTIAL_DADS View 55
   3.24 The QUADS View 59
   3.25 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views 60
   3.26 The SWERB view 62
   3.27 The SWERB_DEPARTS view 64
   3.28 The SWERB_LOC_GPS_XY view 64
   3.29 The SWERB_LOCs view 65
   3.30 The SWERB_UPLOAD view 66
   3.31 The ULNA_STATS View 66
   3.32 The TESTES_ARC_STATS View 69
   3.33 The TESTES_DIAM_STATS View 72
## 4 Views Which Add Gid To Tables

4.1 The BIRTH_GRP View

4.2 The ENTRYDATE_GRP View

4.3 The STATDATE_GRP View

4.4 The CONSORTDATES_GRP View

4.5 The CYCGAPDAYS_GRP View

4.6 The CYCGAPS_GRP View

4.7 The CYCSTATS_GRP View

4.8 The DARTINGS_GRP View

4.9 The DISPERSEDATES_GRP View

4.10 The MATUREDATES_GRP View

4.11 The MDINTERVALS_GRP View

4.12 The MMINTERVALS_GRP View

4.13 The PCSKINS_GRP View

4.14 The RANKDATES_GRP View

4.15 The REPSTATS_GRP View
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 Hybrid Teeth and Ticks Entity and Relationship Diagram ......................... 15
12. Babase Hybrid Scores Data Entity Relationship Diagram ...................................... 16
13. Babase SWERB Core Tables Entity Relationship Diagram .................................. 17
14. Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram .... 18
15. Babase Manual Weather Data Entity Relationship Diagram .................................. 19
16. Babase WeatherHawk Data Entity Relationship Diagram ..................................... 20
17. Warning Sub-System Entity Relationship Diagram ............................................. 21
18. Query Defining the ACTOR_ACTEES View ........................................................... 22
19. Entity Relationship Diagram of the ACTOR_ACTEES View .................................. 23
20. Query Defining the ANESTH_STATS View ......................................................... 23
21. Entity Relationship Diagram of the ANESTH_STATS View ................................... 24
22. Query Defining the BODYTEMP_STATS View ..................................................... 24
23. Entity Relationship Diagram of the BODYTEMP_STATS View .............................. 25
24. Query Defining the CENSUS_DEMOG View ....................................................... 25
25. Entity Relationship Diagram of the CENSUS_DEMOG View ............................... 25
26. Query Defining the CHEST_STATS View ............................................................. 26
27. Entity Relationship Diagram of the CHEST_STATS View ..................................... 26
28. Query Defining the CROWNRUMP_STATS View ............................................... 27
29. Entity Relationship Diagram of the CROWNRUMP_STATS View ........................ 27
30. Query Defining the CYCLES_SEXSKINS View ................................................... 27
31. Entity Relationship Diagram of the CYCLES_SEXSKINS View ............................ 28
32. Query Defining the CYCPOINTS_CYCLES View ............................................... 28
33. Entity Relationship Diagram of the CYCPOINTS_CYCLES View ........................ 28
34. Query Defining the DSAMPLES View ................................................................. 30
35. Query Defining the DEMOG_CENSUS View ...................................................... 31
36. Entity Relationship Diagram of the DEMOG_CENSUS View ............................... 31
37. Query Defining the DENT_CODES View ............................................................. 33
38 Entity Relationship Diagram of the DENT_CODES View .................................................. 34
39 Query Defining the DENT_SITES View ................................................................. 36
40 Entity Relationship Diagram of the DENT_SITES View ........................................... 37
41 Query Defining the INTERACT View ................................................................. 38
42 Entity Relationship Diagram of the INTERACT View .............................................. 38
43 Query Defining the MATERNITIES View ............................................................. 39
44 Entity Relationship Diagram of the MATERNITIES View ....................................... 40
45 Query Defining the MIN_MAXS View ................................................................. 41
46 Entity Relationship Diagram of the MIN_MAXS View ............................................ 42
47 Query Defining the MIN_MAXS_SORTED View ...................................................... 43
48 Entity Relationship Diagram of the MIN_MAXS_SORTED View .................................. 44
49 Query Defining the MPI_EVENTS View .............................................................. 46
50 Entity Relationship Diagram of the MPI_EVENTS View .......................................... 47
51 Query Defining the MTD_CYCLES View ............................................................. 48
52 Entity Relationship Diagram of the MTD_CYCLES View ........................................ 49
53 Query Defining the PARENTS View ................................................................. 50
54 Entity Relationship Diagram of the PARENTS View .............................................. 51
55 Query Defining the PCSKINS_SORTED View ....................................................... 51
56 Entity Relationship Diagram of the PCSKINS_SORTED View ................................... 52
57 Query Defining the PCV_STATS View ............................................................... 52
58 Entity Relationship Diagram of the PCV_STATS View ........................................... 52
59 Query Defining the POINTS View ................................................................. 53
60 Entity Relationship Diagram of the POINTS View ............................................... 53
61 Query Defining the POTENTIAL_DADS View ...................................................... 55
62 Entity Relationship Diagram of the foundation of the POTENTIAL_DADS View ........ 56
63 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 ............................................. 57
64 Entity Relationship Diagram of that portion of the POTENTIAL_DADS View having easily computed columns ................................................................. 58
65 Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions ................................................................. 59
66 Query Defining the QUADS View ................................................................. 59
67 Entity Relationship Diagram of the QUADS View ............................................... 60
68 Query Defining the SESSKINS_CYCLES View ...................................................... 60
69 Entity Relationship Diagram of the SESSKINS_CYCLES View ................................ 60
70 Query Defining the SWEB View ........................................................................... 62
71 Entity Relationship Diagram of the SWEB View .................................................... 63
72 Query Defining the SWEB_DEPARTS View .......................................................... 64
73 Entity Relationship Diagram of the SWEB_DEPARTS View .................................. 64
74 Query Defining the SWEB_LOC_GPS XY View ...................................................... 64
75 Entity Relationship Diagram of the SWEB_LOC_GPS XY View .............................. 65
76 Query Defining the SWERB_LOCS View .......................................................... 65
77 Entity Relationship Diagram of the SWERB_LOCS View ............................... 65
78 Query Defining the SWERB_UPLOAD View ...................................................... 66
79 Entity Relationship Diagram of the SWERB_UPLOAD View ............................ 66
80 Query Defining the ULNA_STATS View ......................................................... 66
81 Entity Relationship Diagram of the ULNA_STATS View .................................. 67
82 Query Defining the TESTES_ARC_STATS View ............................................ 69
83 Entity Relationship Diagram of the TESTES_ARC_STATS View ....................... 70
84 Query Defining the TESTES_DIAM_STATS View .......................................... 72
85 Entity Relationship Diagram of the TESTES_DIAM_STATS View ....................... 73
86 Query Defining the BIRTH_GRP View ......................................................... 74
87 Entity Relationship Diagram of the BIRTH_GRP View .................................. 74
88 Query Defining the ENTRYDATE_GRP View .................................................. 74
89 Entity Relationship Diagram of the ENTRYDATE_GRP View ........................... 74
90 Query Defining the STATDATE_GRP View .................................................... 74
91 Entity Relationship Diagram of the STATDATE_GRP View ............................. 75
92 Query Defining the CONSORTDATES_GRP View .......................................... 75
93 Entity Relationship Diagram of the CONSORTDATES_GRP View ....................... 75
94 Query Defining the CYCGAPDAYS_GRP View ............................................. 75
95 Entity Relationship Diagram of the CYCGAPDAYS_GRP View ........................ 75
96 Query Defining the CYCGAPS_GRP View .................................................... 76
97 Entity Relationship Diagram of the CYCGAPS_GRP View ................................ 76
98 Query Defining the CYCSTATS_GRP View ................................................... 76
99 Entity Relationship Diagram of the CYCSTATS_GRP View ................................ 76
100 Query Defining the DARTINGS_GRP View .................................................. 77
101 Entity Relationship Diagram of the DARTINGS_GRP View ............................. 77
102 Query Defining the DISPERSEDATES_GRP View ....................................... 77
103 Entity Relationship Diagram of the DISPERSEDATES_GRP View .................... 77
104 Query Defining the MATUREDATES_GRP View .......................................... 78
105 Entity Relationship Diagram of the MATUREDATES_GRP View ......................... 78
106 Query Defining the MDINTERV ALS_GRP View ......................................... 78
107 Entity Relationship Diagram of the MDINTERV ALS_GRP View ....................... 78
108 Query Defining the MMINTERV ALS_GRP View ......................................... 79
109 Entity Relationship Diagram of the MMINTERV ALS_GRP View ....................... 79
110 Query Defining the PCSKINS_GRP View .................................................... 79
111 Entity Relationship Diagram of the PCSKINS_GRP View ............................. 79
112 Query Defining the RANKDATES_GRP View ............................................... 80
113 Entity Relationship Diagram of the RANKDATES_GRP View .......................... 80
114 Query Defining the REPSTATS_GRP View ............................................... 80
115 Entity Relationship Diagram of the REPSTATS_GRP View ........................... 80
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 BABAЕ Database

Figure 1: Key to the Babase Entity Relationship Diagrams

1At this time of this writing only males have data entered into RANKDATES in Babase.
### Group Membership and Life Events

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTERNATE_SNAMES in Babase:</td>
<td>rescinded name</td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>animal, including fetuses</td>
</tr>
<tr>
<td>CENSUS in Babase:</td>
<td>day each individual is (or is not) observed in a group</td>
</tr>
<tr>
<td>CONSORTDATES in Babase:</td>
<td>male who has a known first consortship</td>
</tr>
<tr>
<td>DEMOG in Babase:</td>
<td>mention of an individual’s presence in a group within a field textual note</td>
</tr>
<tr>
<td>DISPERSEDATES in Babase:</td>
<td>male who has left his maternal study group</td>
</tr>
<tr>
<td>GROUPS in Babase:</td>
<td>group (including solitary males)</td>
</tr>
<tr>
<td>MATUREDATES in Babase:</td>
<td>individual who is sexually mature</td>
</tr>
<tr>
<td>RANKDATES in Babase:</td>
<td>individual(^1) who has attained adult rank</td>
</tr>
</tbody>
</table>

### Analyzed: Group Membership and Life Events

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>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>RANKS in Babase:</td>
<td>month each individual is ranked in each group</td>
</tr>
</tbody>
</table>

### Sexual Cycles

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</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>PCSKINS in Babase:</td>
<td>PCS color 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>sexskin measurement of each female</td>
</tr>
</tbody>
</table>

### The Sexual Cycle Day-By-Day Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYCGAPDAYS in Babase:</td>
<td>female for each day within a period during which there is not continuous observation</td>
</tr>
<tr>
<td>CYCGAPDAYS in Babase:</td>
<td>female for each day within a period during which there is not continuous observation</td>
</tr>
<tr>
<td>CYCGAPDAYS in Babase:</td>
<td>female for each day within a period during which there is not continuous observation</td>
</tr>
<tr>
<td>CYCSTATS in Babase:</td>
<td>day each female is cycling -- by M, T and Ddates</td>
</tr>
<tr>
<td>MDINTERVALS in Babase:</td>
<td>day each female is cycling and is between M and Ddates</td>
</tr>
<tr>
<td>MMINTERVALS in Babase:</td>
<td>day each female is cycling -- by Mdates</td>
</tr>
<tr>
<td>REPSTATS in Babase:</td>
<td>day each female has a known reproductive state</td>
</tr>
</tbody>
</table>

### Social and Multiparty Interactions

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLMISCS in Babase:</td>
<td>‘free form’ all-occurrences datum</td>
</tr>
<tr>
<td>CONSORTS in Babase:</td>
<td>multiparty dispute over a consortship</td>
</tr>
<tr>
<td>FPOINTS in Babase:</td>
<td>point observation of a mature female</td>
</tr>
<tr>
<td>INTERACT_DATA in Babase:</td>
<td>interaction between individuals</td>
</tr>
<tr>
<td>MPIS in Babase:</td>
<td>collection of multiparty interactions</td>
</tr>
</tbody>
</table>
### Table 2: The Warning Sub-System Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGRITY QUERIES in Babase:</td>
<td>query used to discover data integrity problems</td>
<td></td>
</tr>
<tr>
<td>INTEGRITY WARNINGS in Babase:</td>
<td>data integrity problem discovered by the warning sub-system</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2: Babase Group Membership Entity Relationship Diagram**

- **BIOGRAPH**: Censuses and other events that place individual in a group.
- **MEMBERS**: All groups an individual is a member of on a day.
- **GROUPS**: Daily membership of the group.
- **CENSUS**: Group of an individual's censusing.
- **DEMOG**: Demography note of placement.
- **Origin**
- **Date**
- **Gid**
- **Name**
- **From_group**
- **Permanent**
- **Supergroup**
- **Start**
- **Cease_To_Exist**
- **Three_letter_code**
- **One_letter_code**
- **Last_Reg_Census**
- **Study_Grp**
| General Support Tables                                                                                                                                                                                                                                                                                                                                                   |
|---|---|---|---|
| **Table** | **Id Column** | **Related Column(s)** | **One entry for every possible choice of...** |
| **OBSERVERS in Babase:** | Initials | **SAMPLES** in Babase: Observer in Babase: **WREADINGS** in Babase: **WRperson** in Babase: **RGSETUPS** in Babase: **RGSPerson** in Babase: **CROWNJRUMP** in Babase: **CRowserver** in Babase: **CHESTS** in Babase: **Choobserver** in Babase: **ULNAS** in Babase: **Ulobserver** in Babase: **HUMERUSES** in Babase: **Huoobserver** in Babase: **SWERB_OBSERVERS** in Babase: Observer in Babase: | person who records information |
| **OBSERVER_ROLES in Babase:** | Initials | **OBSERVERS** in Babase: **Role** in Babase: **OBSERVERS** in Babase: **SWERB_ObsERVER_Role** in Babase: **OBSERVERS** in Babase: **SWERB_Driver_Role** in Babase: **SWERB_OBSERVERS** in Babase: **Role** in Babase: | role in which a person can be involved in the data collection process |
| **UNKSNAMES in Babase:** | Unksname | **NEIGHBORS** in Babase: Unksname in Babase: and the **SWERB_UPLOAD** in Babase: view | problem in identifying neighbor of focal during point sampling or in identifying a lone male in a SWERB other group observation |
| **Group Membership and Life Events**                                                                                                                                                                                                                                                                                                                                 |   |
| **Table** | **Id Column** | **Related Column(s)** | **One entry for every possible choice of...** |
| **BSTATUSES in Babase:** | Bstatus | **BIOGRAPH** in Babase: Bstatus in Babase: | birthday estimation accuracy |
| **CONFIDENCES in Babase:** | Confidence | **BIOGRAPH** in Babase: **DcauseNatureConfidence** in Babase: **BIOGRAPH** in Babase: **DcauseAgentConfidence** in Babase: **DISPERSEDATES** in Babase: **Disconfidence in Babase:** **BIOGRAPH** in Babase: **Matgrconfidencen in Babase:** | degree of certitude in nature of death, agent of death, disperse date assignment, or maternal group assignment |
| **DAD_SOFTWARE in Babase:** | Software | **DAD_DATA** in Babase: Software in Babase: | software package used to perform genetic paternity analysis |
| **DCAUSES in Babase:** | Dcause | **BIOGRAPH** in Babase: Dcause in Babase: | cause of death |
| **DEATHNATURES in Babase:** | Nature | **DCAUSES** in Babase: Nature in Babase: | reason for death |
| **DEMOG_REFERENCES in Babase:** | Reference | **DEMOG** in Babase: Reference in Babase: | data source for demography notes |
| **MSTATUSES in Babase:** | Mstatus | **Babase:** Matured in Babase: **RANKDATES** in Babase: **Ranked in Babase:** | maturity marker date estimation process |
Table 4: The Warning Sub-System Support Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Id Column</th>
<th>Related Column(s)</th>
<th>One entry for every possible choice of...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQTYPES in Babase:</td>
<td>IQType</td>
<td>INTEGRITY QUERIES in Babase:</td>
<td>kind of problem with data integrity</td>
</tr>
<tr>
<td>WARNING_REMARKS in Babase:</td>
<td>WRID</td>
<td>INTEGRITY WARNINGS in Babase:</td>
<td>remark which might apply to more than one instance of questionable database integrity</td>
</tr>
</tbody>
</table>

Table 4: The Warning Sub-System Support Tables

Figure 3: Babase Life Events Entity Relationship Diagram
## Group Membership and Life Events

<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-NULLDad_consensus in Babase:.</td>
<td>Easy access to parental information.</td>
<td>BIOGRAPH in Babase:, MATERNITIES in Babase:, DAD_DATA in Babase:, MEMBERS in Babase:</td>
</tr>
<tr>
<td>POTENTIAL_DADS in Babase:</td>
<td></td>
<td>(completed) female reproductive event for every male more than 2192 days old (approximately 6 years) present in the mother’s group during her fertile period</td>
<td>MATERNITIES in Babase:, MEMBERS in Babase: (multiple times), ACTOR_ACTEES in Babase: (multiple times), BIOGRAPH in Babase:, RANKDATES in Babase:, MATUREDATES in Babase:</td>
</tr>
</tbody>
</table>

## Sexual Cycles

<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>CYCLES_SEXSKINS in Babase:</td>
<td>CYCLES in Babase: row</td>
<td>Maintenance of SEXSKINS in Babase: rows.</td>
<td>CYCLES in Babase:, SEXSKINS in Babase:</td>
</tr>
<tr>
<td>Table</td>
<td>View</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>BIOGRAPH</code> in <code>Babase</code></td>
<td><code>BIRTH_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>BIOGRAPH</code> in <code>Babase</code></td>
<td><code>ENTRYDATE_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>BIOGRAPH</code> in <code>Babase</code></td>
<td><code>STATDATE_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>CONSORTDATES</code> in <code>Babase</code></td>
<td><code>CONSORTDATES_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>CYCGAPDAYS</code> in <code>Babase</code></td>
<td><code>CYCGAPDAYS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>CYCGAPS</code> in <code>Babase</code></td>
<td><code>CYCGAPS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>CYCSTATS</code> in <code>Babase</code></td>
<td><code>CYCSTATS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>DARTINGS</code> in <code>Babase</code></td>
<td><code>DARTINGS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>DISPERSEDATES</code> in <code>Babase</code></td>
<td><code>DISPERSEDATES_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>MATUREDATES</code> in <code>Babase</code></td>
<td><code>MATUREDATES_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>MDINTERVALS</code> in <code>Babase</code></td>
<td><code>MDINTERVALS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>MMINTERVALS</code> in <code>Babase</code></td>
<td><code>MMINTERVALS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>PCSKINS</code> in <code>Babase</code></td>
<td><code>PCSKINS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>RANKDATES</code> in <code>Babase</code></td>
<td><code>RANKDATES_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>REPSSTATS</code> in <code>Babase</code></td>
<td><code>REPSSTATS_GRP</code> in <code>Babase</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

3.1 The ACTOR_ACTEES View

```
SELECT interact_data.iid AS iid,
       interact_data.sid AS sid,
       interact_data.act AS act,
       interact_data.date AS date,
       interact_data.start AS start,
       interact_data.stop AS stop,
       interact_data.observer AS observer,
       actor.partid AS actorid,
       COALESCE(actor.sname, '998'::CHAR(3)) AS actor,
       (SELECT actorms.grp
        FROM members AS actorms
        WHERE actorms.sname = actor.sname
        AND actorms.date = interact_data.date) AS actor_grp,
       actee.partid AS acteeid,
       COALESCE(actee.sname, '998'::CHAR(3)) AS actee,
       (SELECT acteems.grp
        FROM members AS acteems
        WHERE acteems.sname = actee.sname
        AND acteems.date = interact_data.date) AS actee_grp,
       interact_data.handwritten AS handwritten
FROM interact_data
LEFT OUTER JOIN parts AS actor
  ON (actor.iid = interact_data.iid AND actor.role = 'R')
LEFT OUTER JOIN parts AS actee
  ON (actee.iid = interact_data.iid AND actee.role = 'E');
```

Figure 18: 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 20: Query Defining the ANESTH_STATS View
### 3.3 The BODYTEMP_STATS View

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

Figure 22: Query Defining the BODYTEMP_STATS View
3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views

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

Figure 24: Query Defining the CENSUS_DEMOG View
3.5 The CHEST_STATS View

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

Figure 26: Query Defining the CHEST_STATS View

Figure 27: Entity Relationship Diagram of the CHEST_STATS View
3.6 The CROWNRUMP_STATS View

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

Figure 28: 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

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

Figure 30: Query Defining the CYCLES_SEXSKINS View
3.8 The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views

```sql
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 32: Query Defining the CYCPOINTS_CYCLES View

Figure 31: Entity Relationship Diagram of the CYCLES_SESKINS View
3.9 The DSAMPLES View

```sql
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
```

Figure 34: Query Defining the DSAMPLES View
### 3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views

The following query defines the DEMOG_CENSUS view:

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

**Figure 35: Query Defining the DEMOG_CENSUS View**

Figure 36 shows the entity relationship diagram of the DEMOG_CENSUS view:

**Figure 36: Entity Relationship Diagram of the DEMOG_CENSUS View**
3.11 The DENT_CODES View

SELECT teethdartids.dartid AS dartid
, rum3.rum3tstate AS rum3tstate
, rum3.rum3tcondition AS rum3tcondition
, rum2.rum2tstate AS rum2tstate
, rum2.rum2tcondition AS rum2tcondition
, rum1.rum1tstate AS rum1tstate
, rum1.rum1tcondition AS rum1tcondition
, rup2.rup2tstate AS rup2tstate
, rup2.rup2tcondition AS rup2tcondition
, rup1.rup1tstate AS rup1tstate
, rup1.rup1tcondition AS rup1tcondition
, ruc.ructstate AS ructstate
, ruc.ructcondition AS ructcondition
, rui2.rui2tstate AS rui2tstate
, rui2.rui2tcondition AS rui2tcondition
, rui1.rui1tstate AS rui1tstate
, rui1.rui1tcondition AS rui1tcondition
, lui1.lui1tstate AS lui1tstate
, lui1.lui1tcondition AS lui1tcondition
, lui2.lui2tstate AS lui2tstate
, lui2.lui2tcondition AS lui2tcondition
, luc.luctstate AS luctstate
, luc.luctcondition AS luctcondition
, lup1.lup1tstate AS lup1tstate
, lup1.lup1tcondition AS lup1tcondition
, lup2.lup2tstate AS lup2tstate
, lup2.lup2tcondition AS lup2tcondition
, lum1.lum1tstate AS lum1tstate
, lum1.lum1tcondition AS lum1tcondition
, lum2.lum2tstate AS lum2tstate
, lum2.lum2tcondition AS lum2tcondition
, lum3.lum3tstate AS lum3tstate
, lum3.lum3tcondition AS lum3tcondition
, llm3.llm3tstate AS llm3tstate
, llm3.llm3tcondition AS llm3tcondition
, llm2.llm2tstate AS llm2tstate
, llm2.llm2tcondition AS llm2tcondition
, llm1.llm1tstate AS llm1tstate
, llm1.llm1tcondition AS llm1tcondition
, llp2.llp2tstate AS llp2tstate
, llp2.llp2tcondition AS llp2tcondition
, llp1.llp1tstate AS llp1tstate
, llp1.llp1tcondition AS llp1tcondition
, llc.llctstate AS llctstate
, llc.llctcondition AS llctcondition
, ltd.ltdtstate AS ltdtstate
, ltd.ltdtcondition AS ltdtcondition
, rdr.rdrcondition AS rdrcondition
, rdr.rdrstate AS rdrstate
;
Figure 38: Entity Relationship Diagram of the DENT_CODES View

The Tstate value of the TEETH row having the correct Darlid and a Tooth value corresponding to the Tooth code appearing in the column name, or NULL if no such row exists.

(TCtcondition) +

The Tcondition value of the TEETH row having the correct Darlid and a Tooth value corresponding to the Tooth code appearing in the column name, or NULL if no such row exists.

(Darlid) +

The unique Darlid values occurring in TEETH.

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

```sql
SELECT teethdartids.dartid AS dartid,
    s1.s1tstate AS s1tstate,
    s1.s1deciduous AS s1deciduous,
    s2.s2tstate AS s2tstate,
    s2.s2deciduous AS s2deciduous,
    s3.s3tstate AS s3tstate,
    s3.s3deciduous AS s3deciduous,
    s4.s4tstate AS s4tstate,
    s4.s4condition AS s4condition,
    s4.s4deciduous AS s4deciduous,
    s5.s5tstate AS s5tstate,
    s5.s5condition AS s5condition,
    s5.s5deciduous AS s5deciduous,
    s6.s6tstate AS s6tstate,
    s6.s6condition AS s6condition,
    s6.s6deciduous AS s6deciduous,
    s7.s7tstate AS s7tstate,
    s7.s7condition AS s7condition,
    s7.s7deciduous AS s7deciduous,
    s8.s8tstate AS s8tstate,
    s8.s8condition AS s8condition,
    s8.s8deciduous AS s8deciduous,
    s9.s9tstate AS s9tstate,
    s9.s9condition AS s9condition,
    s9.s9deciduous AS s9deciduous,
    s10.s10tstate AS s10tstate,
    s10.s10condition AS s10condition,
    s10.s10deciduous AS s10deciduous,
    s11.s11tstate AS s11tstate,
    s11.s11condition AS s11condition,
    s11.s11deciduous AS s11deciduous,
    s12.s12tstate AS s12tstate,
    s12.s12condition AS s12condition,
    s12.s12deciduous AS s12deciduous,
    s13.s13tstate AS s13tstate,
    s13.s13condition AS s13condition,
    s13.s13deciduous AS s13deciduous,
    s14.s14tstate AS s14tstate,
    s14.s14condition AS s14condition,
    s14.s14deciduous AS s14deciduous,
    s15.s15tstate AS s15tstate,
    s15.s15condition AS s15condition,
    s15.s15deciduous AS s15deciduous,
    s16.s16tstate AS s16tstate,
    s16.s16condition AS s16condition,
    s16.s16deciduous AS s16deciduous,
    s17.s17tstate AS s17tstate,
    s17.s17condition AS s17condition,
    s17.s17deciduous AS s17deciduous,
    s18.s18tstate AS s18tstate,
    s18.s18condition AS s18condition,
    s18.s18deciduous AS s18deciduous,
    s19.s19tstate AS s19tstate,
    s19.s19condition AS s19condition,
    s19.s19deciduous AS s19deciduous,
    s20.s20tstate AS s20tstate,
    s20.s20condition AS s20condition
FROM toothcodes, teeth
WHERE toothcodes.toothsite = '32'
LEFT OUTER JOIN teeth AS s29
    ON s29.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s28
    ON s28.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s27
    ON s27.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s26
    ON s26.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s25
    ON s25.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s24
    ON s24.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s23
    ON s23.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s22
    ON s22.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s19
    ON s19.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s18
    ON s18.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s16
    ON s16.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s15
    ON s15.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s12
    ON s12.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s10
    ON s10.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s8
    ON s8.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s6
    ON s6.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s4
    ON s4.dartid = teeth.dartid
LEFT OUTER JOIN teeth AS s2
    ON s2.dartid = teeth.dartid
WHERE toothcodes.toothsite = '29'
LEFT OUTER JOIN toothcodes AS s32
    ON s32.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s31
    ON s31.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s30
    ON s30.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s29
    ON s29.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s28
    ON s28.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s27
    ON s27.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s26
    ON s26.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s25
    ON s25.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s24
    ON s24.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s23
    ON s23.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s22
    ON s22.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s19
    ON s19.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s18
    ON s18.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s16
    ON s16.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s15
    ON s15.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s12
    ON s12.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s10
    ON s10.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s8
    ON s8.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s6
    ON s6.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s4
    ON s4.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s2
    ON s2.dartid = teeth.dartid
WHERE toothcodes.toothsite = '27'
LEFT OUTER JOIN toothcodes AS s22
    ON s22.dartid = teeth.dartid
LEFT OUTER JOIN toothcodes AS s21
    ON s21.dartid = teeth.dartid
WHERE toothcodes.toothsite = '24'
LEFT OUTER JOIN toothcodes AS s19
    ON s19.dartid = teeth.dartid
WHERE toothcodes.toothsite = '22'
LEFT OUTER JOIN toothcodes AS s18
    ON s18.dartid = teeth.dartid
WHERE toothcodes.toothsite = '21'
LEFT OUTER JOIN toothcodes AS s16
    ON s16.dartid = teeth.dartid
WHERE toothcodes.toothsite = '20'
LEFT OUTER JOIN toothcodes AS s14
    ON s14.dartid = teeth.dartid
WHERE toothcodes.toothsite = '19'
LEFT OUTER JOIN toothcodes AS s12
    ON s12.dartid = teeth.dartid
WHERE toothcodes.toothsite = '18'
LEFT OUTER JOIN toothcodes AS s10
    ON s10.dartid = teeth.dartid
WHERE toothcodes.toothsite = '17'
LEFT OUTER JOIN toothcodes AS s8
    ON s8.dartid = teeth.dartid
WHERE toothcodes.toothsite = '16'
LEFT OUTER JOIN toothcodes AS s6
    ON s6.dartid = teeth.dartid
WHERE toothcodes.toothsite = '15'
LEFT OUTER JOIN toothcodes AS s4
    ON s4.dartid = teeth.dartid
WHERE toothcodes.toothsite = '14'
LEFT OUTER JOIN toothcodes AS s2
    ON s2.dartid = teeth.dartid
WHERE toothcodes.toothsite = '13'
LEFT OUTER JOIN toothcodes AS s1
    ON s1.dartid = teeth.dartid
WHERE toothcodes.toothsite = '12'
WHERE toothcodes.toothsite = '11'
WHERE toothcodes.toothsite = '10'
WHERE toothcodes.toothsite = '9'
WHERE toothcodes.toothsite = '8'
WHERE toothcodes.toothsite = '7'
WHERE toothcodes.toothsite = '6'
WHERE toothcodes.toothsite = '4'
WHERE toothcodes.toothsite = '3'
WHERE toothcodes.toothsite = '2'`
Figure 40: Entity Relationship Diagram of the DENT_SITES View
3.13 The INTERACT and INTERACT_SORTED Views

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

Figure 41: Query Defining the INTERACT View

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

Figure 42: Entity Relationship Diagram of the INTERACT View
### 3.14 The MATERNITIES View

```sql
SELECT cycles.sname AS mom,
      cycles.cid AS cid,
      cycles.seq AS seq,
      cycles.series AS series,
      cycpoints.cpid AS conceive,
      cycpoints.date AS zdate,
      members.grp AS zdate_grp,
      cycpoints.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 43: Query Defining the MATERNITIES View
Figure 44: Entity Relationship Diagram of the MATERNITIES View

Although a join on this column alone returns multiple rows, because there is another join on a different column only 1 row matches all the criteria. (The combination of Sname and Date is unique.)
3.15 The MIN_MAXS View

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

Figure 45: Query Defining the MIN_MAXS View
Figure 46: Entity Relationship Diagram of the MIN_MAXS View
3.16 The MIN_MAXS_SORTED View

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

Figure 47: Query Defining the MIN_MAXS_SORTED View
Figure 48: Entity Relationship Diagram of the MIN_MAXS_SORTED View
### 3.17 The MPI_EVENTS View

```sql
SELECT mpis.mpiid AS mpiid,
       mpis.date AS date,
       mpis.context_type AS context_type,
       mpis.context AS context,
       mpi_data.mpidid AS mpidid,
       mpi_data.seq AS seq,
       mpi_data.mpiact AS mpiact,
       actor.mpipid AS actorid,
       actor.sname AS actor,
       actor.unksname AS unkactor,
       actee.mpipid AS acteeid,
       actee.sname AS actee,
       actee.unksname AS unkactee,
       CASE WHEN EXISTS(SELECT 1
                       FROM mpiacts
                       WHERE mpiacts.mpiact = mpi_data.mpiact
                             AND mpiacts.kind = 'H')
              THEN
              EXISTS(SELECT 1
                      FROM mpi_data AS request,
                           mpiacts,
                           mpi_parts AS requestor,
                           mpi_parts AS requestee
                      WHERE request.mpiid = mpi_data.mpiid
                            AND request.seq < mpi_data.seq
                            AND mpiacts.mpiact = request.mpiact
                            AND mpiacts.kind = 'R'
                            AND requestor.mpidid = request.mpidid
                            AND requestor.role = 'R'
                            AND requestor.sname = actee.sname
                            AND requestee.mpidid = request.mpidid
                            AND requestee.role = 'E'
                            AND requestee.sname = actor.sname)
              ELSE
              NULL
              END AS solicited,
       EXISTS(SELECT 1
                   FROM mpi_data AS initial,
                        mpiacts
                   WHERE initial.mpiid = mpi_data.mpiid
                         AND initial.seq = 1
                         AND mpiacts.mpiact = initial.mpiact
                         AND mpiacts.decided)
       AS decided,
       mpi_data.helped AS helped,
       mpi_data.active AS active
FROM mpis
LEFT OUTER JOIN mpi_data ON (mpis.mpiid = mpi_data.mpiid)
LEFT OUTER JOIN mpi_parts AS actor
    ON (actor.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 49: Query Defining the MPI_EVENTS View
Figure 50: Entity Relationship Diagram of the MPI_EVENTS View
### 3.18 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 dsourc
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 51: Query Defining the MTD_CYCLES View
Figure 52: Entity Relationship Diagram of the MTD_CYCLES View
3.19 The PARENTS View

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

Figure 53: Query Defining the PARENTS View
3.20 The PCSKINS_SORTED View

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

Figure 55: Query Defining the PCSKINS_SORTED View
3.21 The PCV_STATS View

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

Figure 57: Query Defining the PCV_STATS View

Figure 58: Entity Relationship Diagram of the PCV_STATS View
### 3.22 The POINTS and POINTS_SORTED Views

```sql
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 59: Query Defining the POINTS View

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

Figure 60: Entity Relationship Diagram of the POINTS View
3.23 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 supergroup(mommembers.grp, mommembers.date)
      = supergroup(dadmembers.grp, dadmembers.date))
WHERE dadmembers.sname = pdads.sname
  AND dadmembers.date < maternities.zdate
  AND dadmembers.date >= maternities.zdate - 5
  AND mommembers.sname = maternities.mom
  AND mommembers.date < maternities.zdate
  AND mommembers.date >= maternities.zdate - 5)
AS estrous_presence
, (SELECT count(*)
FROM actor_actees
WHERE actor_actees.date < maternities.zdate
  AND actor_actees.date >= maternities.zdate - 5
  AND (actor_actees.act = 'M'
      OR actor_actees.act = 'E')
  AND actor_actees.actor = pdads.sname
  AND actor_actees.actee = maternities.mom)
AS estrous_me
, (SELECT count(*)
FROM actor_actees
WHERE actor_actees.date < maternities.zdate
  AND actor_actees.date >= maternities.zdate - 5
  AND actor_actees.act = 'C'
  AND actor_actees.actor = pdads.sname
  AND actor_actees.actee = maternities.mom)
AS estrous_c

FROM maternities
JOIN biograph AS pdads
  ON (pdads.sname
      IN (SELECT dadmembers.sname
            FROM members AS dadmembers
            JOIN members AS mommembers
              ON (mommembers.date = dadmembers.date
                  AND supergroup(mommembers.grp, mommembers.date)
                  = supergroup(dadmembers.grp, dadmembers.date))
            WHERE dadmembers.sname = pdads.sname
                  AND dadmembers.date < maternities.zdate
                  AND dadmembers.date >= maternities.zdate - 5
                  AND mommembers.sname = maternities.mom
                  AND mommembers.date < maternities.zdate
                  AND mommembers.date >= maternities.zdate - 5))
LEFT OUTER JOIN rankdates
  ON (rankdates.sname = pdads.sname)
LEFT OUTER JOIN maturedates
  ON (maturedates.sname = pdads.sname)
WHERE pdads.sex = 'M'
-- Speed things up by eliminating potential dads
-- who could not possibly interpolate into the mom's group
-- during the fertile period.
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 62: 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 63: 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.
* PDADS is an alias for BIOGRAPH, representing those BIOGRAPH rows that satisfy the conditions required to be considered a potential dad of a given kid. It does not appear anywhere as an independent entity. Additional conditions, as shown on other diagrams, must also be true for a BIOGRAPH row to be a PDADS row.

Figure 64: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View having easily computed columns
3.24 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 66: Query Defining the QUADS View
3.25 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views

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

Figure 68: Query Defining the SEXSKINS_CYCLES View
3.26 The SWERB view

```
SELECT swerb_data.swid AS swid
, swerb_departs_data.did AS did
, swerb_departs_data.date AS date
, swerb_data.time AS time
, swerb_bes.beid AS beid
, swerb_bes.focal_grp AS focal_grp
, swerb_bes.seq AS seq
, swerb_data.event AS event
, swerb_data.seen_grp AS seen_grp
, swerb_data.lone_animal AS lone_animal
, swerb_data.quad AS quad
, CASE
  WHEN swerb_data.xyloc IS NULL
  THEN 'quad'
  ELSE 'gps'
END AS merged_is
, 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_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.esource AS esource
, swerb_bes.is_effort AS is_effort
, swerb_departs_gps.gps AS gps
, swerb_bes.notes AS notes
FROM swerb_data
LEFT OUTER JOIN quad_data
  ON (quad_data.quad = swerb_data.quad)
JOIN swerb_bes
  ON (swerb_bes.beid = swerb_data.beid)
JOIN swerb_departs_data
  ON (swerb_departs_data.did = swerb_bes.did)
LEFT OUTER JOIN swerb_departs_gps
  ON (swerb_departs_gps.did = swerb_bes.did)
LEFT OUTER JOIN swerb_loc_data
  ON (swerb_loc_data.swid = swerb_data.swid)
LEFT OUTER JOIN adcodes
  ON (adcodes.adcode = swerb_loc_data.adcode)
LEFT OUTER JOIN swerb_loc_gps
```

Figure 70: Query Defining the SWERB View
Figure 71: Entity Relationship Diagram of the SWERB View
3.27 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 72: Query Defining the SWERB_DEPARTS View

3.28 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 74: Query Defining the SWERB_LOC_GPS_XY View
3.29 The SWERB_LOCS view

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

Figure 77: Entity Relationship Diagram of the SWERB_LOCS View
3.30  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::BOOLEAN AS secondary_ad,
       NULL::TEXT AS notes
WHERE _raise_babase_exception('Cannot select SWERB_UPLOAD'
                             || ': The only use of the SWERB_UPLOAD view is to insert'
                             || ' new data into the SWERB portion of babase');
```

Figure 78: 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 79: Entity Relationship Diagram of the SWERB_UPLOAD View

3.31  The ULNA_STATS View

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

Figure 80: Query Defining the ULNA_STATS View
Figure 81: Entity Relationship Diagram of the ULNA_STATS View
3.32 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 83: Entity Relationship Diagram of the TESTES_ARC_STATS View
3.33 The TESTES_DIAM_STATS View

SELECT testesdartids.dartid AS dartid,
  , testesllength.testllengthsamps AS testllengthsamps
  , testesllength.testllength_mean AS testllength_mean
  , testesllength.testllength_stddev AS testllength_stddev
  , testeslwidth.testlwidthsamps AS testlwidthsamps
  , testeslwidth.testlwidth_mean AS testlwidth_mean
  , testeslwidth.testlwidth_stddev AS testlwidth_stddev
  , testesrlength.testrlengthsamps AS testrlengthsamps
  , testesrlength.testrlength_mean AS testrlength_mean
  , testesrlength.testrlength_stddev AS testrlength_stddev
  , testesrwidth.testrwidthsamps AS testrwidthsamps
  , testesrwidth.testrwidth_mean AS testrwidth_mean
  , testesrwidth.testrwidth_stddev AS testrwidth_stddev
FROM (SELECT testes_diam.dartid
    FROM testes_diam
    GROUP BY testes_diam.dartid)
AS testesdartids
LEFT OUTER JOIN
  (SELECT testes_diam.dartid AS llengthdartid
    , count(*) AS testllengthsamps
    , avg(testes_diam.testlength) AS testllength_mean
    , stddev(testes_diam.testlength) AS testllength_stddev
    FROM testes_diam
    WHERE testes_diam.testside = 'L'
      AND testes_diam.testlength IS NOT NULL
    GROUP BY testes_diam.dartid)
AS testesllength
ON testesllength.llengthdartid = testesdartids.dartid
LEFT OUTER JOIN
  (SELECT testes_diam.dartid AS lwidthdartid
    , count(*) AS testlwidthsamps
    , avg(testes_diam.testwidth) AS testlwidth_mean
    , stddev(testes_diam.testwidth) AS testlwidth_stddev
    FROM testes_diam
    WHERE testes_diam.testside = 'L'
      AND testes_diam.testwidth IS NOT NULL
    GROUP BY testes_diam.dartid)
AS testeslwidth
ON testeslwidth.lwidthdartid = testesdartids.dartid
LEFT OUTER JOIN
  (SELECT testes_diam.dartid AS rlengthdartid
    , count(*) AS testrlengthsamps
    , avg(testes_diam.testlength) AS testrlength_mean
    , stddev(testes_diam.testlength) AS testrlength_stddev
    FROM testes_diam
    WHERE testes_diam.testside = 'R'
      AND testes_diam.testlength IS NOT NULL
    GROUP BY testes_diam.dartid)
AS testesrlength
ON testesrlength.rlengthdartid = testesdartids.dartid
LEFT OUTER JOIN
  (SELECT testes_diam.dartid AS rwidthdartid
    , count(*) AS testrwidthsamps
    , avg(testes_diam.testwidth) AS testrwidth_mean
    , stddev(testes_diam.testwidth) AS testrwidth_stddev
    FROM testes_diam
    WHERE testes_diam.testside = 'R'
      AND testes_diam.testwidth IS NOT NULL
    GROUP BY testes_diam.dartid)
AS testesrwidth
ON testesrwidth.rwidthdartid = testesdartids.dartid;
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 86: 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 (Birth)</td>
<td>Date (Birth)</td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH...</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 87: 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 88: 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 (Entrydate)</td>
<td>Date (Entrydate)</td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH...</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 89: 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 90: Query Defining 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 92: Query Defining 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 94: Query Defining the 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 96: Query Defining the CYCGAPS_GRP View

<table>
<thead>
<tr>
<th>CYCGAPS</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gapid (Gapid)</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 CYCGAPS....</td>
<td></td>
</tr>
</tbody>
</table>

Figure 97: 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 98: 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></td>
</tr>
</tbody>
</table>

Figure 99: 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 100: Query Defining the DARTINGS_GRP View

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

Figure 101: 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 102: Query Defining the DISPERSEDATES_GRP View

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

Figure 103: 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 104: Query Defining the MATUREDATES_GRP View

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

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 106: Query Defining the MDINTERVALS_GRP View

![Entity Relationship Diagram of the MDINTERVALS_GRP View](image)
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 108: 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 109: Entity Relationship Diagram of the MMINTERVALS_GRP View

4.13 The PCSKINS_GRP View

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

Figure 110: Query Defining the PCSKINS_GRP View

<table>
<thead>
<tr>
<th>PCSKINS</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCSId (PCSId)</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 PCSKINS....</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 111: Entity Relationship Diagram of the PCSKINS_GRP View
### 4.14 The RANKDATES_GRP View

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

Figure 112: 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 113: Entity Relationship Diagram of the RANKDATES_GRP View

### 4.15 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 114: 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 115: Entity Relationship Diagram of the REPSTATS_GRP View