**ASSESSMENT 3 – DATABASE SLEUTHING**

**Module:** Colony Module  
**Assumed Database:** MySQL *(adaptable for PostgreSQL if needed)*

**Purpose**

Ensure data integrity in the Colony module (animals, matings, pedigree, cages) by adding automated checks to the CI/CD pipeline. These help detect silent issues before they escalate to production.

**Integrity Check 1: Orphaned Parent Records**

* **What it catches:**  
  Mating records referencing male\_id or female\_id that no longer exist in the animals table (e.g., deleted parent animals).
* **SQL Logic:**

SELECT mating\_id

FROM matings

WHERE male\_id NOT IN (SELECT animal\_id FROM animals)

OR female\_id NOT IN (SELECT animal\_id FROM animals);

* **When it runs:**  
  On every **staging deployment** or **nightly build**.

**Integrity Check 2: Timestamp Reversals**

* **What it catches:**  
  Records where end\_date is earlier than start\_date, especially for cages, litters, or matings.
* **SQL Logic:**

SELECT cage\_id

FROM cages

WHERE end\_date IS NOT NULL AND end\_date < start\_date;

* **When it runs:**  
  On every **pull request** that modifies backend logic or database schema.

**Integrity Check 3: Duplicate Animal IDs or Tag Numbers**

* **What it catches:**  
  Duplicate tag\_number or internal\_id, which must be unique per colony.
* **SQL Logic:**

SELECT tag\_number, COUNT(\*)

FROM animals

GROUP BY tag\_number

HAVING COUNT(\*) > 1;

* **When it runs:**  
  During **CI/CD test suite execution** (e.g., on data migration jobs or nightly builds).

**Optional PostgreSQL Adaptation**

If PostgreSQL is used instead of MySQL, the same checks apply but with enhanced features:

|  |  |
| --- | --- |
| **Feature** | **PostgreSQL Approach** |
| **Foreign Key Handling** | Use FOREIGN KEY constraints with ON DELETE CASCADE |
| **Uniqueness** | Use UNIQUE constraints and EXCLUDE USING for partial uniqueness |
| **Timestamp Logic** | Same as MySQL (direct SQL) |
| **Indexing** | Use **partial indexes** on active records for optimized checks |
| **Auditing** | Optionally integrate pgAudit for tracking changes |

**Conclusion**

These database integrity checks are critical for:

* Maintaining valid, reliable Colony data
* Catching backend bugs or schema regressions early
* Ensuring test and production environments stay clean and trusted