

Database Design Document

Animal Shelter Management System

Final Project - INF 305: Database Management Systems 2

TABLE OF CONTENTS

1. System Overview
 2. Entity Relationship Diagram (ERD)
 3. Normalization Analysis
 4. Table Descriptions
 5. Constraints & Relationships
 6. Data Dictionary
 7. Design Decisions
-

1. SYSTEM OVERVIEW

Project Purpose

The Animal Shelter Management System is a comprehensive database and web application that manages: - Multiple animal shelters/facilities - 150+ animals with complete tracking - Veterinary care and medical records - Adoption and donor tracking - Staff management and scheduling

System Goals

- Eliminate data redundancy through normalization
 - Ensure data integrity through constraints
 - Optimize query performance through views
 - Encapsulate business logic in PL/SQL
 - Provide secure web interface via Oracle APEX
-

2. ENTITY RELATIONSHIP DIAGRAM (ERD)

Core Entities and Relationships

SHELTERS (1) ----< (M) ANIMALS
|
+-----< (M) STAFF
|
+-----< (M) DONATIONS

```

ANIMALS (1) -----< (M) ADOPTIONS
|
+-----< (M) MEDICAL_RECORDS
|
+-----< (M) VACCINATIONS
|
+-----< (M) TREATMENTS
|
+-----< (M) APPOINTMENTS

VETERINARIANS (1) -----< (M) MEDICAL_RECORDS
|
+-----< (M) APPOINTMENTS

ADOPTERS (1) -----< (M) ADOPTIONS

```

Entity Definitions

Entity	Purpose	Key Characteristic
SHELTERS	Facility locations	Multiple facilities managed
ANIMALS	Animals in care	Central entity - 150+ records
VETERINARIANS	Medical professionals	50+ specialists
STAFF	Facility employees	Manage multiple shelters
ADOPTERS	Animal recipients	Track adoption outcomes
ADOPTIONS	Placement records	Complete adoption history
MEDICAL_RECORDS	Health history	Complete medical tracking
VACCINATIONS	Immunization tracking	Schedule management
TREATMENTS	Treatment records	Medical intervention logs
DONATIONS	Financial contributions	Fundraising tracking
APPOINTMENTS	Scheduled visits	Appointment management

3. NORMALIZATION ANALYSIS

Normalization Status: 3NF (Third Normal Form)

1st Normal Form (1NF) - No Repeating Groups

All attributes contain atomic (indivisible) values No repeating groups or arrays in columns Each row is unique

Example: Instead of storing multiple vaccines in one ANIMALS row, vaccines go in separate VACCINATIONS table:

WRONG (violates 1NF):

ANIMALS: animal_id, name, vaccines[]

CORRECT (1NF):

ANIMALS: animal_id, name, ...

VACCINATIONS: vaccination_id, animal_id, vaccine_name, ...

2nd Normal Form (2NF) - Remove Partial Dependencies

Every non-key attribute depends on entire primary key No partial dependencies

Example: Veterinarian name depends on veterinarian_id, not on medical_record_id:

WRONG (violates 2NF):

MEDICAL_RECORDS: medical_id, animal_id, vet_id, vet_name, ...

CORRECT (2NF):

MEDICAL_RECORDS: medical_id, animal_id, vet_id, visit_date, ...

VETERINARIANS: vet_id, vet_name, specialization, ...

3rd Normal Form (3NF) - Remove Transitive Dependencies

No non-key attribute depends on another non-key attribute All non-key attributes depend only on primary key

Example: Shelter name depends on shelter_id, not on animal_id:

WRONG (violates 3NF):

ANIMALS: animal_id, name, shelter_id, shelter_name, ...

CORRECT (3NF):

ANIMALS: animal_id, name, shelter_id, ...

SHELTERS: shelter_id, shelter_name, ...

Benefits Achieved

- Eliminates data redundancy (update shelter once, not 150 times)
- Maintains data integrity (foreign keys ensure consistency)
- Improves query performance (smaller tables, efficient joins)
- Simplifies maintenance (changes propagate automatically)

4. TABLE DESCRIPTIONS

ANIMALS (Primary Table)

Purpose: Store core animal information

Columns:

animal_id (PK)	NUMBER	Primary key, auto-generated by sequence
shelter_id (FK)	NUMBER	References SHELTERS, identifies location
name	VARCHAR2(100)	Animal name (required)
species	VARCHAR2(50)	Type (Dog, Cat, Rabbit, etc.)
breed	VARCHAR2(50)	Breed within species
date_of_birth	DATE	Birth date (for age calculation)
gender	CHAR(1)	M = Male, F = Female
color	VARCHAR2(50)	Physical description (Black, Brown, etc.)
weight	NUMBER(5,2)	Weight in kg
intake_date	DATE	When animal arrived at shelter
status	VARCHAR2(20)	Available, Adopted, Fostered, Medical Care, Deceased
microchip_number (UQ)	VARCHAR2(50)	UNIQUE - Critical for lost/found identification
notes	VARCHAR2(500)	Special care requirements or behavioral notes
created_date	DATE	When record created
modified_date	DATE	When record last updated

Constraints: - PRIMARY KEY (animal_id) - FOREIGN KEY (shelter_id) REFERENCES SHELTERS - UNIQUE (microchip_number) - Prevents duplicate microchips - NOT NULL (name, species, breed, shelter_id) - CHECK (gender IN ('M', 'F')) - CHECK (status IN ('Available', 'Adopted', 'Fostered', 'Medical Care', 'Deceased'))

Example Records:

1, 1, Bella, Cat, Siamese, 06-JUN-2022, F, Cream, 4.5, 01-JAN-2023, Available, MC#001, Playful
 2, 1, Max, Dog, Labrador, 15-AUG-2021, M, Golden, 28.3, 15-MAY-2023, Available, MC#002, Friendly

SHELTERS

Purpose: Store facility location and contact information

Columns:

shelter_id (PK)	NUMBER	Primary key
name	VARCHAR2(100)	Facility name (Shelter 1-100)
address	VARCHAR2(200)	Physical location
city	VARCHAR2(50)	City name
phone	VARCHAR2(20)	Contact phone
email	VARCHAR2(100)	Contact email
capacity	NUMBER	Max animals facility can hold

Constraints: PRIMARY KEY, NOT NULL on name

Example: (1, 'Shelter 1', '123 Main St', 'New York', '212-555-0100', 'contact@shelter1.org', 100)

VETERINARIANS

Purpose: Store veterinary professional information

Columns:

veterinarian_id (PK)	NUMBER	Primary key
name	VARCHAR2(100)	Vet name
specialization	VARCHAR2(100)	Area of expertise (Small Animals, Large Animals, etc.)
license_number (UQ)	VARCHAR2(50)	License for verification
clinic_name	VARCHAR2(100)	Where they practice
phone	VARCHAR2(20)	Contact phone
email	VARCHAR2(100)	Contact email

Constraints: PRIMARY KEY, UNIQUE (license_number), NOT NULL on required fields

STAFF

Purpose: Store employee information and facility assignments

Columns:

staff_id (PK)	NUMBER	Primary key
shelter_id (FK)	NUMBER	References SHELTERS
name	VARCHAR2(100)	Employee name
position	VARCHAR2(50)	Job title (Manager, Handler, Cleaner)
email	VARCHAR2(100)	Work email
phone	VARCHAR2(20)	Work phone
hire_date	DATE	Employment start date

Constraints: PRIMARY KEY, FOREIGN KEY (shelter_id), NOT NULL on required fields

ADOPTERS

Purpose: Store information about people adopting animals

Columns:

adopter_id (PK)	NUMBER	Primary key
name	VARCHAR2(100)	Adopter name
email	VARCHAR2(100)	Contact email
phone	VARCHAR2(20)	Contact phone
address	VARCHAR2(200)	Home address
city	VARCHAR2(50)	City
state	VARCHAR2(2)	State/Province
zip_code	VARCHAR2(10)	Postal code

adoption_count NUMBER How many animals adopted

Constraints: PRIMARY KEY, NOT NULL on required fields

ADOPTIONS

Purpose: Track adoption transactions linking animals to adopters

Columns:

adoption_id (PK)	NUMBER	Primary key
animal_id (FK)	NUMBER	References ANIMALS
adopter_id (FK)	NUMBER	References ADOPTERS
adoption_date	DATE	When adoption completed
status	VARCHAR2(20)	Pending, Approved, Completed, Returned
follow_up_date	DATE	When to check on animal
notes	VARCHAR2(500)	Special conditions of adoption

Constraints: PRIMARY KEY, FOREIGN KEYs, NOT NULL on required fields

Purpose: Links ANIMALS to ADOPTERS in many-to-many relationship

MEDICAL_RECORDS

Purpose: Store complete medical history for animals

Columns:

medical_id (PK)	NUMBER	Primary key
animal_id (FK)	NUMBER	References ANIMALS
veterinarian_id (FK)	NUMBER	References VETERINARIANS
visit_date	DATE	When visit occurred
diagnosis	VARCHAR2(200)	What was diagnosed
treatment	VARCHAR2(500)	What treatment given
prescription	VARCHAR2(500)	Medications prescribed
follow_up_needed	CHAR(1)	Y/N - Whether follow-up needed
follow_up_date	DATE	When follow-up scheduled
notes	VARCHAR2(500)	Additional medical notes

Constraints: PRIMARY KEYs, FOREIGN KEYs, NOT NULL on required fields

VACCINATIONS

Purpose: Track immunization records and schedules

Columns:

vaccination_id (PK)	NUMBER	Primary key
animal_id (FK)	NUMBER	References ANIMALS
vaccine_name	VARCHAR2(100)	Type of vaccine
vaccination_date	DATE	When given
next_due_date	DATE	When booster needed
veterinarian_id (FK)	NUMBER	Who administered
notes	VARCHAR2(500)	Vaccine batch number, reaction notes

Constraints: PRIMARY KEYs, FOREIGN KEYs

TREATMENTS

Purpose: Store treatment and therapy records

Columns:

treatment_id (PK)	NUMBER	Primary key
animal_id (FK)	NUMBER	References ANIMALS
treatment_type	VARCHAR2(100)	Type (Physical therapy, Surgery, etc.)
treatment_date	DATE	When treatment given
description	VARCHAR2(500)	What was done
cost	NUMBER(8,2)	Treatment cost
completed	CHAR(1)	Y/N - Is treatment finished
notes	VARCHAR2(500)	Additional notes

Constraints: PRIMARY KEYs, FOREIGN KEYs

DONATIONS

Purpose: Track financial contributions from donors

Columns:

donation_id (PK)	NUMBER	Primary key
shelter_id (FK)	NUMBER	References SHELTERS (which shelter got donation)
donor_name	VARCHAR2(100)	Name of donor
donor_email	VARCHAR2(100)	Email for thank you
amount	NUMBER(10,2)	Donation amount
donation_date	DATE	When received
donation_type	VARCHAR2(50)	Cash, Check, Credit Card, In-Kind
notes	VARCHAR2(500)	Special designation/purpose
tax_receipt_sent	CHAR(1)	Y/N - Receipt mailed

Constraints: PRIMARY KEY, FOREIGN KEY, NOT NULL on required fields

APPOINTMENTS

Purpose: Schedule veterinary appointments

Columns:

appointment_id (PK)	NUMBER	Primary key
animal_id (FK)	NUMBER	References ANIMALS
veterinarian_id (FK)	NUMBER	References VETERINARIANS
appointment_date	DATE	When appointment scheduled
appointment_time	VARCHAR2(5)	Time (HH:MM format)
type	VARCHAR2(50)	Checkup, Vaccination, Surgery, etc.
reason	VARCHAR2(200)	Why appointment needed
completed	CHAR(1)	Y/N - Did appointment happen
notes	VARCHAR2(500)	Notes about appointment

Constraints: PRIMARY KEYs, FOREIGN KEYs

5. CONSTRAINTS & RELATIONSHIPS

Primary Keys

Every table has primary key for unique identification: - Prevents duplicate rows
- Enables efficient lookups - Required for foreign key relationships

Foreign Keys

Connect related tables:

```
ANIMALS.shelter_id → SHELTERS.shelter_id
ADOPTIONS.animal_id → ANIMALS.animal_id
ADOPTIONS.adopter_id → ADOPTERS.adopter_id
MEDICAL_RECORDS.animal_id → ANIMALS.animal_id
MEDICAL_RECORDS.veterinarian_id → VETERINARIANS.veterinarian_id
VACCINATIONS.animal_id → ANIMALS.animal_id
TREATMENTS.animal_id → ANIMALS.animal_id
APPOINTMENTS.animal_id → ANIMALS.animal_id
APPOINTMENTS.veterinarian_id → VETERINARIANS.veterinarian_id
STAFF.shelter_id → SHELTERS.shelter_id
DONATIONS.shelter_id → SHELTERS.shelter_id
```

Benefits of Foreign Keys: - Prevent orphaned records (can't delete shelter if animals exist) - Ensure referential integrity (can't add animal with non-existent shelter) - Enable cascade operations (delete shelter → delete animals)

Unique Constraints

Prevent duplicates in specific fields: - ANIMALS.microchip_number - Each animal has unique chip - VETERINARIANS.license_number - Each vet has unique license

Check Constraints

Validate data values: - ANIMALS.gender IN ('M', 'F') - Only male or female - ANIMALS.status IN valid statuses - Prevents invalid states - DONATIONS.amount > 0 - No negative donations

NOT NULL Constraints

Enforce required fields: - animal_id, shelter_id, name, species, breed (in ANIMALS) - Ensures completeness of critical data

6. DATA DICTIONARY

Core Statistics

Metric	Value
Total Tables	11
Primary Keys	11
Foreign Keys	10
Unique Constraints	2
Check Constraints	5+
NOT NULL Constraints	30+
Sequences	3
Views	10

Record Counts (Sample Data)

Table	Records
SHELTERS	10
ANIMALS	150+
VETERINARIANS	50+
STAFF	25
ADOPTERS	30+
ADOPTIONS	30+
MEDICAL_RECORDS	80+
VACCINATIONS	120+
TREATMENTS	50+

Table	Records
DONATIONS	40+
APPOINTMENTS	25+

Data Types Used

Type	Purpose	Examples
NUMBER	Numeric data	IDs, prices, counts
NUMBER(5,2)	Decimal values	Weight (25.50 kg)
VARCHAR2(n)	Variable text	Names, descriptions
DATE	Calendar dates	Birth dates, adoption dates
CHAR(1)	Fixed single character	Gender (M/F), Status flags

7. DESIGN DECISIONS

Why 11 Tables?

Each table has a specific purpose: 1. **ANIMALS** - Core entity (150+ records) 2. **SHELTERS** - Multi-facility support 3. **VETERINARIANS** - Medical professional separation 4. **STAFF** - Employee management 5. **ADOPTERS** - Adoption tracking 6. **ADOPTIONS** - Junction table (many-to-many: animals adopters) 7. **MEDICAL_RECORDS** - Complete medical history 8. **VACCINATIONS** - Immunization tracking 9. **TREATMENTS** - Treatment/therapy records 10. **DONATIONS** - Financial tracking 11. **APPOINTMENTS** - Scheduling

Why This Normalization?

- **3NF is optimal for transactional systems**
- Eliminates redundancy (update shelter name once)
- Maintains consistency (foreign keys)
- Enables efficient queries (focused tables)
- Supports complex relationships (medical history)

Alternative Considerations (NOT Used)

- Could denormalize for reporting → Used views instead
- Could split ANIMALS further → Would over-complicate
- Could combine related tables → Would violate normalization

Sequences vs Auto-Increment

Chosen: Sequences (`seq_animal.NEXTVAL`) - Atomic, no gaps (mostly) - Works across distributed systems - Oracle standard practice - Can be controlled programmatically

Triggers for Consistency

Automatic updates on certain operations: - Update `ANIMALS.status` when adoption created - Log timestamp on record modifications - Cascade delete related records safely

Views for Performance

Instead of complex queries in APEX: - `v_animal_summary` - Pre-joined animal + shelter + age - `v_adoptions_month` - Pre-aggregated monthly adoptions - Simplifies application code - Centralizes query logic - Can be indexed for speed

CONCLUSION

This database design achieves: - Data integrity through constraints - Efficiency through normalization - Flexibility through relationships - Consistency through foreign keys - Performance through views

The 11-table, 3NF design supports managing 150+ animals across multiple shelters with complete medical, adoption, and financial tracking.

Document Version: 1.0 **Created:** December 2025 **Course:** INF 305 - Database Management Systems 2