

Progetto Basi di Dati

“Parco Zoologico con Clinica Veterinaria”

di

Fenu Lucia
1125521

Sacchetto Francesco
1136168

1. Abstract

Uno giardino zoologico, più comunemente chiamato Zoo, è il luogo dove vengono esposti al pubblico varie specie di animali. Questi animali possono essere relativamente comuni, ma anche esotici e ricercati, spaziando nei vari generi. Generalmente uno Zoo viene considerato un luogo ricreativo, ma non va dimenticata la sua importantissima funzione di conservazione della biodiversità. Vi possono essere, ad esempio, degli animali a rischio di estinzione, che altrimenti risulterebbero in pericolo nei loro luoghi d'origine. Ogni animale è inserito in una gabbia (o recinto), che cerca di replicare il suo habitat naturale, rispettando le sue esigenze. Inoltre, spesso uno Zoo di medie-grandi dimensioni risulta avere al suo interno una Clinica Veterinaria, ove è presente un team di veterinari specializzati, atta alla salvaguardia e cura degli animali presenti.

Sarà dunque necessario poter disporre di un efficiente metodo di catalogazione dei dipendenti dello Zoo (tra cui i veterinari), degli animali presenti e del loro quadro clinico. Si potrà avere accesso a tutto ciò che riguarda l'animale, che rappresenta la principale risorsa da proteggere e curare. Per ciascuno di essi sarà possibile infatti ottenere informazioni che spaziano dal lato gestionale (gabbia, addetti assegnati, etc) al lato clinico.

2. Analisi dei requisiti

Si vuole realizzare una base di dati per la gestione di uno Zoo con una Clinica Veterinaria al suo interno. Lo scopo è quello di organizzare al meglio ogni aspetto della realtà di interesse, ponendo l'attenzione sia sul lato gestionale che medico. Lo **Zoo** ha: un **codice identificativo**, un **nome** e un **indirizzo**. Tale informazione però non è di particolare importanza per la gestione degli animali e dei dipendenti, poiché si presume che questa base di dati sia gestita da un unico Zoo. Ogni **animale** possiede un **nome**, **data di nascita** e **sex**; presenta caratteristiche specifiche come la **nazione di provenienza**, l'**habitat**, la **famiglia** e la **specie**. E' munito inoltre di un **codice identificativo** unico e di un **codice gabbia** in cui è collocato; quest'ultima identificata a sua volta anche dal **nome** e dall'**habitat** che riproduce ed emula. Tutti i **dipendenti** che ruotano attorno alla gestione e cura degli animali vengono catalogati e identificati mediante i loro dati anagrafici (**nome**, **cognome** e **codice fiscale**) e da un **numero di matricola** personale. Per dipendente si intendono due tipi di figura professionale: l'**addetto** a una specifica gabbia e il **Medico Veterinario**. L'addetto alla gabbia è identificato attraverso la **mansione** che svolge (manutenzione, riparazioni, pulizia, approvvigionamenti etc.) e il codice della gabbia a cui è associato. Il veterinario invece risulta essere una figura più complessa, non solo dotata di una **specializzazione**, ma soprattutto per il ruolo che svolge. Un veterinario ha la possibilità di stilare la **cartella clinica** di uno o più animali. Tale cartella è resa unica grazie al codice dell'animale associato e contiene dati generici come la **dieta** che segue ed eventuali caratteristiche o **segni particolari**. Un veterinario può anche eseguire delle **visite**, che verranno poi catalogate per andare a creare quello che è comunemente chiamato lo storico clinico. Dunque, ogni visita sarà munita di un **codice identificativo** affiancato al codice dell'animale visitato. I dati all'interno della stessa, saranno il codice del veterinario (la sua matricola), la **data** in cui è stata effettuata e ovviamente le **motivazioni** per cui è stata richiesta. Queste motivazioni possono riguardare una semplice vaccinazione o dei veri e propri sintomi patologici. In caso di vaccinazione, quindi, risulterà necessario tenere traccia dei **vaccini** eseguiti, attraverso il **codice**, il **nome del farmaco**, il **principio attivo** e la **data** di iniezione. Nel caso in cui si tratti di una motivazione patologica, sarà opportuno avere dei record contenenti sia i **sintomi** che presenta l'animale, sia la **diagnosi** con annessa terapia. Se la terapia presenta dei **farmaci**, allora si terrà traccia di essi, come già visto con i vaccini; registrando il loro **nome**, **codice** e **principio attivo**. Tutto ciò rappresenta una semplice catalogazione di tutto ciò che riguarda e ruota attorno all'animale, ma è importante anche poter disporre di interrogazioni specifiche alla suddetta base di dati. Sarà utile infatti poter risalire, attraverso il codice dell'animale, a tutti i suoi dati. Se si cercano informazioni riguardanti la parte gestionale, sarà possibile vedere in che gabbia è collocato e di conseguenza sapere chi se ne occupa e con quale mansione. Parlando poi della parte clinica, sempre attraverso il suo codice identificativo, è possibile risalire alla sua cartella generale e a tutte le visite veterinarie a cui è stato sottoposto, con annessi vaccini ed eventuali terapie (con farmaci e non).

3. Progettazione Concettuale

3.1 Entità, Attributi e Gerarchie

- **Zoo:** Modella le informazioni relative al giardino zoologico. Questa entità non è tuttavia strettamente legata agli animali, in quanto si assume che tutta la gestione dei dati avvenga già direttamente all'interno dello stesso. (Si rimanda alla sezione 4 per ulteriori spiegazioni.)

Attributi:

- CodiceZoo: Int (Identificatore)
- Nome: String
- Indirizzo (Attributo composto):
 - Via: String
 - NCivico: Int
 - Città: String
 - Cap: Int

- **Animale:** Modella l'entità più importante della realtà di interesse, in quanto tutto il lavoro di gestione, manutenzione e cura è incentrato su essi.

Attributi:

- CodiceAnimale: Int (Identificatore)
- CodiceGabbia: Int (Identificatore)
- CodiceZoo: Int (Identificatore)
- Informazioni (Attributo composto):
 - Nome: String
 - DataNascita: Date
 - Sesso: Char
 - Habitat: String
 - NazioneProvenienza: String
 - Famiglia: String
 - Specie: String

- **Gabbia:** Descrive il luogo di collocazione dell'animale.

Attributi:

- CodiceGabbia: Int (Identificatore)
- NomeGabbia: String
- Habitat: String

- **Dipendente:** Modella la figura lavorativa all'interno dello Zoo. I dipendenti che ruotano attorno all'animale si occupano essenzialmente di due aspetti: gestione/manutenzione e cura. Per il primo vi sono gli addetti alle gabbie, che si occupano di pulizie, rifornimenti etc.... Per la cura, il compito è affidato a Medici Veterinari specializzati che lavorano all'interno della clinica dello Zoo.

Attributi:

- Matricola: Int (Identificatore)
- Nome: String
- Cognome: String
- CodiceFiscale: String

Gerarchia – Entità figlie:

Addetto: Svolge varie mansioni tra cui la manutenzione, pulizia e riparazione della gabbia, e rifornimento di cibo.

Attributi:

- Mansione: String
- CodiceGabbia: String

Veterinario: Rappresenta la figura lavorativa con più responsabilità e oneri, poiché si occupa della cura e del benessere dell'animale, che talvolta può essere raro, in via d'estinzione o comunque minacciato.

Attributi:

- Specializzazione: String

- **Cartella Clinica**: Modella le informazioni relative alla salute dell'animale. Si tratta di una cartella molto generale in cui per "Segni particolari" si intendono tutte quelle informazioni utili a distinguere l'animale sotto un quadro sanitario. Ad esempio, vengono raccolti dati su eventuali allergie a farmaci e/o sostanze e/o cibi.
Attributi:
-CodiceAnimale: Int (Identificatore)
-Dieta: String
-SegniParticolari: String
- **Visita**: A differenza della Cartella Clinica dell'animale che raccoglie solo le informazioni di base, le visite rappresentano il vero aspetto clinico dell'animale. Dalla visita infatti ci si può diramare verso le cure e i vaccini e gli eventuali farmaci sotto cui l'animale è in cura.
Attributi:
-CodiceVisita: Int (identificatore)
-CodiceAnimale: Int
-Data: Date
-Motivazione: String
-CodiceVeterinario: Int (Identificatore)
- **Diagnosi**: Una visita può comportare a una diagnosi, sia per motivi patologici che per necessità d'intervento. Questa entità ne modella gli aspetti principali attraverso stringhe con le informazioni redatte dal Medico Veterinario.
Attributi:
-CodiceVisita: Int (Identificatore)
-Sintomi: String
-Diagnosi: String
-Terapia: String
-CodiceFarmaco: Int
- **Farmaco**: Modella le informazioni del Farmaco somministrato all'animale. Vengono riportati sia il codice del farmaco, ma anche il nome dello stesso, per facilitarne la lettura complessiva.
Attributi:
-CodiceFarmaco: Int (Identificatore)
-NomeFarmaco: String (Identificatore)
-PrincipioAttivo: String
- **Vaccino**: Si è scelto di rappresentare l'entità "Vaccino" a sé stante e non come generalizzazione di "Farmaco". Questo perché il farmaco è legato a una visita di carattere patologico e/o di incombenza necessaria, mentre il vaccino è obbligatorio a priori per l'animale. Seguiranno note nell'analisi delle ridondanze.
Attributi:
-CodiceFarmaco: Int (Identificatore)
-NomeFarmaco: String (Identificatore)
-CodiceVisita: Int (Identificatore)
-PrincipioAttivo: String

3.2 Relazioni e Cardinalità

- **Gabbia-Animale: Collocazione**
-(0,N): In una gabbia possono essere collocati uno o più animali, a seconda della necessità. Può trattarsi di esemplari diversi della stessa specie o anche di specie diverse, purché venga rispettato il micro ecosistema. Può succedere però che una gabbia sia temporaneamente vuota e/o in allestimento.
-(1,1): Ogni animale viene collocato in una gabbia
- **Animale-Zoo: Appartenenza**
-(1,1): Ogni animale appartiene a uno e un solo Zoo. Può succedere che un animale provenga da uno Zoo differente e sia stato semplicemente "prestato" per, ad esempio, cure specifiche o per motivi di riproduzione.
-(1,N): Uno Zoo può possedere uno o (quasi sicuramente) più animali.

- **Zoo-Dipendente: Occupazione**

-(1,N): All'interno di uno Zoo lavorano da uno a molti dipendenti, ciascuno con la propria mansione.

-(1,1): Un dipendente lavora sola nello Zoo in cui è assunto.

- **Veterinaria-Cartella Clinica: Redazione**

-(0,N): Un Medico Veterinario può redigere la cartella clinica di uno o più animali. Tuttavia, possono esserci Veterinari che si occupano solo dell'ambito clinico e, dunque, non redigono le cartelle cliniche.

-(1,1): Ogni cartella clinica viene redatta da un Veterinario.

- **Veterinario-Visita: Esegue**

-(0,N): Un Veterinario esegue delle visite agli animali. Poiché vi sono Veterinari assunti per le loro specializzazioni, può accadere che, se l'animale in particolare non presenta alcuna necessità di intervento, i suddetti non abbiano ancora eseguito alcuna visita.

-(1,1) Ogni visita viene eseguita da un Veterinario. Nel caso di una visita di cui si occupino più Veterinari, il Veterinario specializzato per quell'animale figura nel "Codice Veterinario" della visita.

- **Visita-Vaccino: Vaccinazione**

-(0,N): Una visita può avere (oppure no) come motivazione la somministrazione di uno o più vaccini.

-(1,N): Ogni vaccino viene somministrato durante una visita, la cui motivazione è la vaccinazione stessa, però lo stesso vaccino può essere rifatto in un'altra visita, come richiamo.

- **Visita-Diagnosi: Sintomatica**

-(0,1): Una visita può portare a una diagnosi oppure no. Ad esempio, se l'animale mostra comportamenti strani o campanelli d'allarme per disturbi patologici, viene eseguita una visita, ma può succedere che essa sia inconcludente.

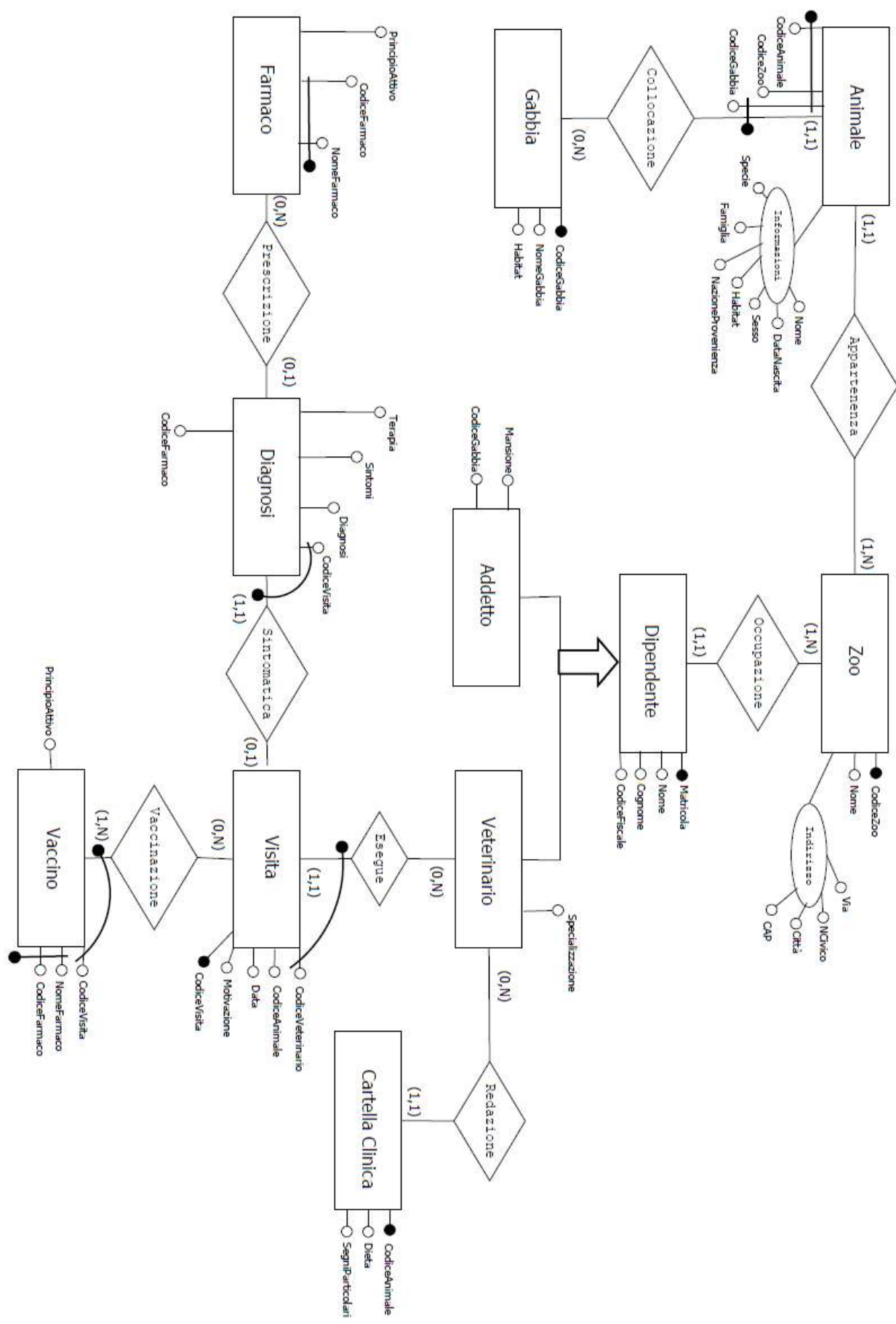
-(1,1): Ogni diagnosi deriva dalla visita appena effettuata.

- **Diagnosi-Farmaco: Prescrizione**

-(0,1): Una diagnosi può comportare la prescrizione di un farmaco. Tuttavia, può accadere che non sia necessario l'uso di sostanze specifiche, ma, ad esempio, un semplice cambio di dieta.

-(0,N): Ogni farmaco a disposizione della clinica viene registrato. Un farmaco viene prescritto dopo la diagnosi di una visita. Uno stesso farmaco può poi venire prescritto più volte, in visite diverse, ad animali diversi.

3.3 Diagramma E-R



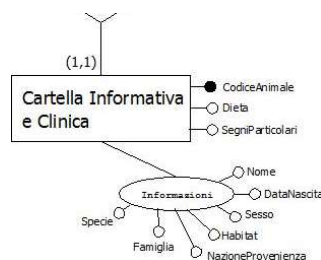
4. Progettazione Logica

4.0 Considerazioni varie

- Questa base di dati ha lo scopo di gestire e organizzare tutti i dati inerenti agli animali, alle gabbie e alla clinica veterinaria. Risulta quindi inusuale avere un'entità "Zoo", poiché è ovvio che questa base di dati viene gestita già all'interno dello stesso. La scelta di introdurre questa entità nasce da un motivo particolare. Spesso accade che uno Zoo possa prestare un suo animale ad un altro Zoo. I motivi possono essere vari: la necessità di cure mediche presso la clinica di un altro Zoo ove vi sia un veterinario specializzato o anche per motivi riproduttivi (mancanza di un partner per un esemplare di una specie a rischio). Per questo motivo ad ogni animale viene assegnato un Codice Zoo che, per la stragrande maggioranza dei casi, sarà lo Zoo stesso della base di dati, ma per alcuni animali potrà essere differente. Per questo occorrerà tenere traccia dei dati (Nome, Codice e Indirizzo) di ciascuno Zoo interessato in questi scambi. Verranno quindi memorizzate anche le informazioni dello Zoo della base di dati, in quanto vi è la possibilità di prestare animali (e non solo riceverli) e le Zoo ricevente avrà bisogno di questo record.
- Per un motivo simile, anche l'entità "Dipendente" va ritoccata. Questa base di dati si occupa della gestione del suo personale e quindi non vi è alcun motivo di mantenere la relazione "Occupazione". Si potrebbe mantenere, inserendo nell'entità "Dipendente" l'attributo "Codice Zoo", ma non ve ne è alcuna necessità perché, a differenza del caso con gli animali, non vi sono dipendenti di Zoo esterni. Viste queste considerazioni, la relazione "Occupazione" viene eliminata poiché superflua.

4.1 Attributi composti

- L'attributo composto "Informazioni" nell'entità "Animale" viene modificato. Nonostante rappresenti dati di carattere informativo, non vi è alcuna necessità di inserirli al momento della registrazione dell'animale nel database. Infatti, all'arrivo (o nascita) di un nuovo animale, occorrerà innanzitutto assegnargli un codice e una collocazione. Solo successivamente verranno inserite tutte le informazioni di genere di quell'animale. Poiché si è notato che questo secondo inserimento avviene in contemporanea con la creazione della "Cartella Clinica" associata, si sceglie dunque di slegare l'attributo composto "Informazioni" dall'entità "Animale" e di inserirlo nell'entità "Cartella Clinica", a cui verrà assegnato il nuovo nome "Cartella Informativa e Clinica".



- L'attributo composto "Indirizzo" nell'entità "Zoo" viene eliminato e ciascuna delle sue componenti viene inserita nella stessa entità.

4.2 Analisi gerarchie

- La gerarchia presente non è completa. L'entità "Dipendente" rappresenta infatti tutte le persone assunte dallo Zoo in questione. L'obiettivo è tuttavia quello di gestire tutti i dati che ruotano attorno all'animale e alla sua salute. In virtù di ciò, figure professionali collegate, ad esempio, all'aspetto amministrativo dello Zoo (uffici, eventi etc.) vengono escluse. L'attenzione viene posta soltanto sulle figure professionali dell'"Addetto" e del "Veterinario". Dunque, per questo tipo di gestione, non vi è la necessità di tenere traccia (o almeno in questa base di dati) di tutti i dipendenti nel loro insieme. L'entità genitore "Dipendente" viene per questo eliminata e tutti i suoi attributi (Matricola e informazioni anagrafiche) vengono ereditati nelle due entità figlie.

L'entità "Addetto" viene dunque scissa (poiché precedentemente si era eliminata la relazione "Occupazione") e collegata logicamente all'entità "Gabbia" attraverso l'attributo comune "Codice Gabbia". (Per l'analisi delle chiavi esterne si veda il paragrafo successivo.)

L'entità "Veterinario", anch'essa scissa, verrà collegata direttamente all'ambito clinico, ossia alla "Cartella Informativa e Clinica" e all'entità "Visita". Inoltre, "Matricola", ora ereditato in "Veterinario" viene rinominato "CodiceVeterinario".

4.3 Chiavi esterne

Nel Diagramma E-R erano presenti molte relazioni concettuali che, ora, verranno sostituite da relazioni logiche attraverso Chiavi Esterne. Per una organizzazione migliore della base di dati, si è preferito creare dei collegamenti come segue:

- **CodiceZoo:**
Chiave esterna tra le entità "Animale" → "Zoo".
- **CodiceGabbia:**
Chiave esterna tra le entità "Animale" → "Gabbia" ← "Addetto".
- **CodiceAnimale:**
Chiave esterna tra le entità "Cartella Informativa e Clinica" → "Animale" ← "Visita".
- **CodiceVeterinario:**
Chiave esterna tra le entità "Veterinario" ← "Visita".
- **CodiceVisita:**
Chiave esterna tra le entità "Diagnosi" → "Visita" ← "Vaccino".
- **CodiceFarmaco:**
Chiave esterna tra le entità "Diagnosi" → "Farmaco".
- **CodiceFarmaco-NomeFarmaco-PrincipioAttivo:**
Chiavi esterne tra le entità "Vaccino" → "Farmaco".

4.4 Analisi ridondanze

All'interno della Clinica Veterinaria è presente l'inventario di tutti i farmaci prescrivibili agli animali. Nella base di dati, questo è modellato dalla tabella dell'entità "Farmaco" che presenta gli attributi "CodiceFarmaco", "NomeFarmaco" e "PrincipioAttivo". Di ogni animale è necessario conoscere lo storico clinico, con particolare attenzione agli ultimi vaccini somministrati. Ogni vaccino è presente nella tabella "Farmaco".

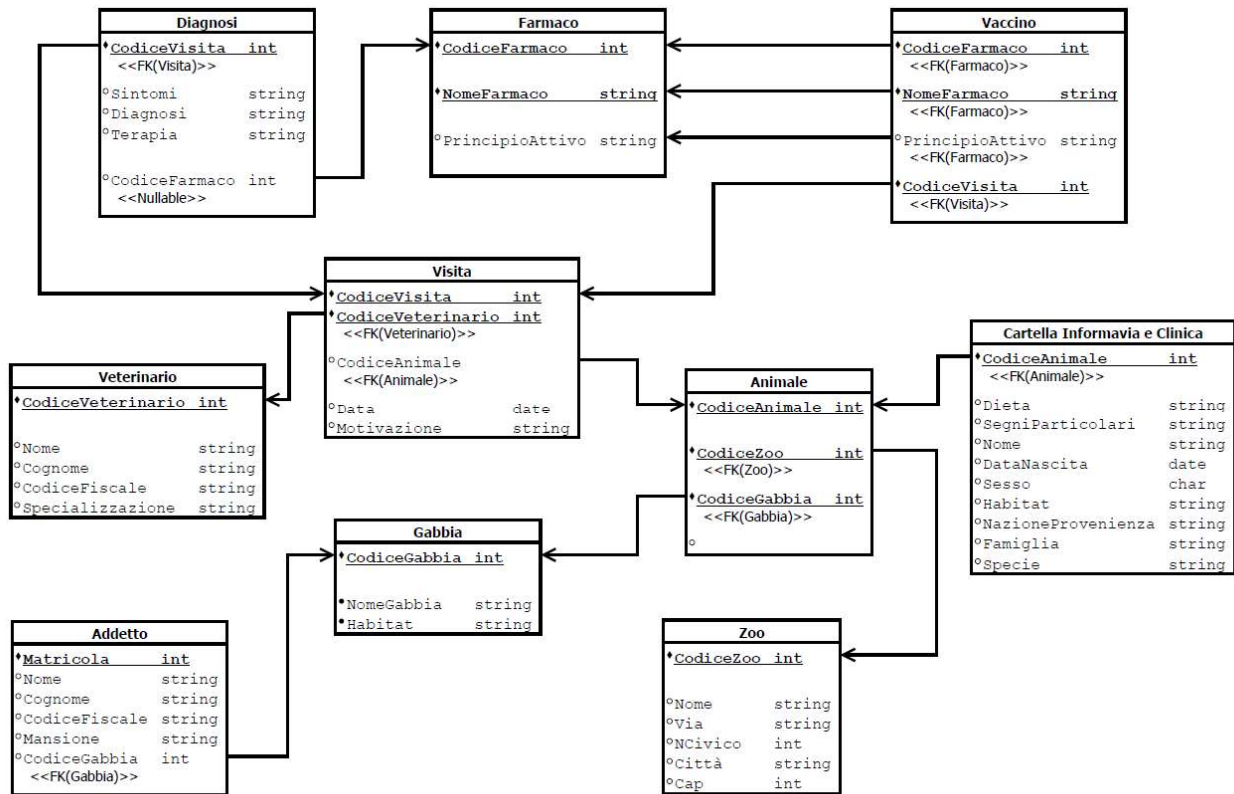
Se si volesse visionare la lista di tutti i vaccini assunti da un animale in particolare, sarebbe necessario inserire un nuovo attributo (che richiami l'animale) nella tabella "Farmaco" per poter effettuare la ricerca.

Tuttavia con la somministrazione di uno stesso vaccino (o farmaco) a due animali diversi, si avrebbe la creazione di due tuple uguali, ma che differiscono soltanto attraverso il codice richiamante l'animale e, oltretutto, senza la possibilità di riconoscere se sia un generico farmaco o vaccino.

Viene quindi a crearsi un ripetersi di informazioni che, ovviamente, aumenta al crescere delle vaccinazioni agli animali dello Zoo. Questa ridondanza, non necessaria per i farmaci generici, lo è invece per poter mantenere lo storico vaccini. La tabella farmaco, come detto in precedenza, rappresenta tuttavia un inventario. Quindi, per sapere quando e quali vaccinazioni sono state effettuate ad un animale in particolare, è più opportuno creare una tabella "Vaccino" atta a tale scopo, per rendere la base di dati più fluente a livello gestionale per ogni animale.

Queste considerazioni hanno portato al mantenimento dell'entità "Vaccino", a cui sono stati aggiunti ulteriori attributi come appunto la "Data" e il "CodiceVisita". Attraverso un semplicissimo controllo incrociato attraverso il "CodiceAnimale" e il "CodiceVisita", sarà possibile visualizzare la lista dei vaccini eseguiti al singolo animale, con la possibilità di fare updates singoli in caso di richiami. Inoltre, in questo modo, è possibile mantenere la differenza tra farmaci generici e vaccini, che risulta essere molto utile in caso di controlli ad ampio spettro sulle vaccinazioni all'interno dello Zoo, che, per legge, sono obbligatorie. L'ipotetica ridondanza di informazioni quali i Codici, Nomi e Principi attivi nella tabella "Farmaco" viene quindi evitata, lasciando per essa soltanto la funzione di inventario; viene invece mantenuta all'interno della tabella "Vaccino" per motivi concettuali e gestionali.

4.5 Schema Logico



4.6 Schema Relazionale

- Entità:

ANIMALE(CodiceAnimale, CodiceZoo, CodiceGabbia)

ZOO(CodiceZoo, Nome, Via, NCivico, Città, Cap)

GABBIA(CodiceGabbia, NomeGabbia, Habitat)

ADDETTO(Matricola, Nome, Cognome, CodiceFiscale, Mansione, CodiceGabbia)

VETERINARIO(CodiceVeterinario, Nome, Cognome, CodiceFiscale, Specializzazione)

CARTELLA_INFORMATIVA_E_CLINICA(CodiceAnimale, Dieta, SegniParticolari, Nome, DataNascita, Sesso, Habitat, NazioneProvenienza, Famiglia, Specie)

VISITA(CodiceVisita, CodiceVeterinario, CodiceAnimale, Data, Motivazione)

DIAGNOSI(CodiceVisita, Sintomi, Diagnosi, Terapia, CodiceFarmaco)

FARMACO(CodiceFarmaco, NomeFarmaco, PrincipioAttivo)

VACCINO(CodiceFarmaco, NomeFarmaco, CodiceVisita, PrincipioAttivo)

- Relazione Molti a Molti:

1. **Vaccinazione**(Visita, Vaccino)

- Relazione Uno a Uno:

1. **Sintomatica**:

(0,1): **VISITA**(CodiceVisita, CodiceVeterinario, CodiceAnimale, Data, Motivazione)

(1,1): **DIAGNOSI**(CodiceVisita, Sintomi, Diagnosi, Terapia, CodiceFarmaco, **VISITA**)

- Relazioni Uno a Molti:

1. Collocazione:

(0,N): GABBIA(CodiceGabbia, NomeGabbia, Habitat)

(1,1): ANIMALE(CodiceAnimale, CodiceZoo, CodiceGabbia, GABBIA)

2. Esegue:

(0,N): VETERINARIO(CodiceVeterinario, Nome, Cognome, CodiceFiscale, Specializzazione)

(1,1): VISITA(CodiceVisita, CodiceVeterinario, CodiceAnimale, Data, Motivazione, VETERINARIO)

3. Prescrizione:

(0,1): DIAGNOSI(CodiceVisita, Sintomi, Diagnosi, Terapia, CodiceFarmaco, NomeFarmaco)

(1,N): FARMACO(CodiceFarmaco, NomeFarmaco, PrincipioAttivo)

Complessivamente:

ANIMALE(CodiceAnimale, CodiceZoo, CodiceGabbia, GABBIA)

ZOO(CodiceZoo, Nome, Via, NCivico, Città, Cap)

GABBIA(CodiceGabbia, NomeGabbia, Habitat)

ADDETTO(Matricola, Nome, Cognome, CodiceFiscale, Mansione, CodiceGabbia)

VETERINARIO(CodiceVeterinario, Nome, Cognome, CodiceFiscale, Specializzazione)

CARTELLA_INFORMATIVA_E_CLINICA(CodiceAnimale, Dieta, SegniParticolari, Nome, DataNascita, Sesso, Habitat, NazioneProvenienza, Famiglia, Specie)

VISITA(CodiceVisita, CodiceVeterinario, CodiceAnimale, Data, Motivazione, VETERINARIO)

DIAGNOSI(CodiceVisita, Sintomi, Diagnosi, Terapia, CodiceFarmaco, NomeFarmaco)

FARMACO(CodiceFarmaco, NomeFarmaco, PrincipioAttivo)

VACCINO(CodiceFarmaco, NomeFarmaco, CodiceVisita, PrincipioAttivo)

Vaccinazione(Visita, Vaccino)

5. Funzioni, Query e Trigger

5.1 Funzioni

a) Si vuole creare una funzione che passato come parametro il codice di un animale, restituisca la sua età (espressa in anni).

```
DELIMITER $$
CREATE FUNCTION `Eta`(codAnim INTEGER(10))
RETURNS int(11)
BEGIN
DECLARE nascita DATE;
DECLARE dataCor DATE;
DECLARE anni INTEGER;
DECLARE giorni INTEGER;
SELECT
    DataNascita
INTO nascita FROM
    Cartella AS c
WHERE
    codAnim = c.CodiceAnimale;
SELECT CURDATE() INTO dataCor;
SELECT DATEDIFF(dataCor, nascita) INTO giorni;
SET anni = giorni/365;
RETURN anni;
END$$
```

b) Si vuole creare una funzione che passati come parametro in ingresso il codice di un farmaco e il codice di un animale, restituisca quante volte gli è stato somministrato.

```
DELIMITER $$
FUNCTION `Usi`(codFarm INTEGER(10), codAnim INTEGER(10))
RETURNS int(11)
BEGIN
DECLARE nVac INTEGER;
DECLARE nDia INTEGER;
DECLARE nUsi INTEGER;
SELECT
    COUNT(*)
INTO nVac FROM
    Visita AS vi,
    Vaccino AS va
WHERE
    codAnim = vi.CodiceAnimale
    AND vi.CodiceVisita = va.CodiceVisita
    AND codFarm = va.CodiceFarmaco;
SELECT
    COUNT(*)
INTO nDia FROM
    Visita AS vi,
    Diagnosi AS d
WHERE
    codAnim = vi.CodiceAnimale
    AND vi.CodiceVisita = d.CodiceVisita
    AND codFarm = d.CodiceFarmaco;
SET nUsi = nVac+nDia;
RETURN nUsi;
END$$
DELIMITER ;
```

5.2 Trigger

a) Si vuole creare un trigger che controlli se la diagnosi, inserita dopo una visita, sia permanente. Se così fosse, la permanenza della suddetta risulterebbe essere inerente alla descrizione dell'animale: va inserita dunque all'interno dei "Segni Particolari" nella sua "Cartella".

```
DROP TRIGGER IF EXISTS UpdatePermanenti;
DELIMITER $$
CREATE TRIGGER UpdatePermanenti AFTER INSERT ON `Diagnosi` FOR EACH ROW
BEGIN
    DECLARE dia VARCHAR(250);
    DECLARE segn VARCHAR(300);
    DECLARE codAn INTEGER;

    IF (new.Diagnosi LIKE 'Permanente%')
    THEN
        SELECT Visita.CodiceAnimale
        INTO codAn FROM
            Visita
        WHERE
            new.CodiceVisita = Visita.CodiceVisita;
        SELECT
            Cartella.SegniParticolari
        INTO segn FROM
            Cartella
        WHERE
            codAn = Cartella.CodiceAnimale;
        UPDATE Cartella
        SET
            SegniParticolari = CONCAT(segn, new.Diagnosi)
        WHERE
            Cartella.CodiceAnimale = codAn;
    END IF;
END$$
DELIMITER;
```

b) Si vuole creare un trigger che prima dell'inserimento in "Diagnosi", controlli, attraverso la funzione "USI()" precedentemente creata, quante volte il farmaco che verrà prescritto è già stato utilizzato. Una volta controllato questo, l'attributo "Terapia" di "Diagnosi" della tupla che si vuole inserire verrà aggiornato aggiungendo la suddetta informazione.

```
DROP TRIGGER IF EXISTS UsiFarmaco;
DELIMITER $$
CREATE TRIGGER UsiFarmaco BEFORE INSERT ON `Diagnosi` FOR EACH ROW
BEGIN
    DECLARE codAnim INTEGER;
    DECLARE usi INTEGER;
    DECLARE testo VARCHAR(150);
    SET testo = 'Uso numero: ';
    SELECT
        Cartella.CodiceAnimale
    INTO codAnim FROM
        Cartella,
        Visita
    WHERE
        Cartella.CodiceAnimale = Visita.CodiceAnimale
        AND Visita.CodiceVisita = new.CodiceVisita;
    SET usi = USI(new.CodiceFarmaco,codAnim) +1;
    SET new.Terapia = CONCAT(new.Terapia,testo,usi,'. ');
END$$
DELIMITER;
```

5.3 Query

a) Una query che restituisca il CodiceAnimale degli animali che hanno ricevuto un vaccino nell'ultimo anno solare, mostrando inoltre in che Data, il nome del Farmaco e da quale veterinario è stato eseguito, (ordinati per CodiceAnimale):

```
SELECT
  a.CodiceAnimale,
  vi.DataVisita as Data, va.NomeFarmaco,
  va.CodiceFarmaco,
  ve.Nome as NomeVeterinario, ve.Cognome as CognomeVeterinario
FROM
  ((Visita AS vi JOIN Animale AS a ON vi.CodiceAnimale = a.CodiceAnimale)
  JOIN Veterinario AS ve ON vi.CodiceVeterinario = ve.CodiceVeterinario)
  JOIN Vaccino AS va ON vi.CodiceVisita = va.CodiceVisita
WHERE
  DATEDIFF(CURDATE(), vi.DataVisita) < 365
ORDER BY a.CodiceAnimale ASC;
```

CodiceAnimale	Data	NomeFarmaco	CodiceFarmaco	NomeVeterinario	CognomeVeterinario
1	2017-03-25	Vaccino per Gastroenterite	2	Beatrice	Poncina
1	2017-03-25	Vaccino AntiTetano.	1	Beatrice	Poncina
1	2017-03-26	Vaccino AntiRabbico	3	Beatrice	Poncina
19	2017-03-25	Vaccino per Gastroenterite	2	Beatrice	Poncina
19	2017-03-25	Vaccino AntiTetano.	1	Beatrice	Poncina
23	2017-03-25	Vaccino per Gastroenterite	2	Beatrice	Poncina
23	2017-03-25	Vaccino AntiTetano.	1	Beatrice	Poncina
23	2017-03-26	Vaccino AntiRabbico	3	Beatrice	Poncina

b) Una query che restituisca il conteggio di tutte le visite sostenute dagli animali, accanto al loro CodiceAnimale e Specie. Per motivi pratici, spesso è più opportuno eseguire la ricerca soltanto per le visite dell'ultimo anno solare.

```
SELECT COUNT(v.CodiceVisita) as Numerovisite, a.CodiceAnimale, c.Specie as Animale
FROM cartella as c, Animale as a, Visita as v
WHERE (CURDATE()-v.DataVisita)<365 AND c.CodiceAnimale=a.CodiceAnimale
      AND a.CodiceAnimale=v.CodiceAnimale
GROUP BY a.CodiceAnimale;
```

Numerovisite	CodiceAnimale	Animale
3	1	Procione
1	3	Volpe
1	5	Puma
1	7	Pinguino Imperatore
1	14	Rinoceronte
1	18	Elefante
1	19	Tigre
1	20	Lemure
1	22	Panda
2	23	Panda Rosso
1	25	Ippopotamo
1	26	Fenicottero rosa

c) Una query che restituisca accanto ad ogni farmaco, il numero di volte in cui è stato somministrato. Per i farmaci, grazie appunto alla tabella "Farmaco", si tiene una sorta di inventario, ma risulta anche utile appunto avere informazioni sugli usi degli stessi.

```
SELECT f.CodiceFarmaco, f.NomeFarmaco, COUNT(f.CodiceFarmaco) as nUSI
FROM Farmaco as f JOIN Diagnosi as d ON (f.CodiceFarmaco=d.CodiceFarmaco)
GROUP BY f.CodiceFarmaco
```

UNION

```
SELECT f.CodiceFarmaco, f.NomeFarmaco, COUNT(f.CodiceFarmaco) as nUSI
FROM Farmaco as f JOIN Vaccino as v ON (f.CodiceFarmaco=v.CodiceFarmaco)
GROUP BY f.CodiceFarmaco
ORDER BY CodiceFarmaco;
```

CodiceFarmaco	NomeFarmaco	nUSI
1	Vaccino AntiTetano.	27
2	Vaccino per Gastroenterite	27
3	Vaccino AntiRabbico	8
4	Antidiarroico	3
6	Antidolorifico generico	1
7	Antiemetico	1
8	Antiacido stomaco	1
9	Antibiotico per infezioni podali	1
11	Calmante	1

d) Poiché solitamente gli animali carnivori sono i più pericolosi, si vuole conoscere il CodiceFiscale e il Cognome degli addetti alle gabbie con animali carnivori di età superiore ai 10 anni. Tali animali, essendo già in età adulta (o comunque non troppo giovani) e avendo convissuto per anni con la presenza umana, risulteranno essere quelli meno pericolosi. Potranno quindi essere gestiti e controllati da Addetti senza troppe precauzioni e/o equipaggiamento protettivo.

```
SELECT a.CodiceAnimale, c.Specie, c.Sesso, ETA(a.CodiceAnimale) as Eta, c.Dieta, a.CodiceGabbia,
ad.CodiceFiscale, ad.Cognome
FROM Cartella AS C, Animale AS a, Addetto AS ad
WHERE C.Dieta = 'Carnivoro' AND ETA(a.CodiceAnimale)>10
AND a.CodiceAnimale = c.CodiceAnimale
AND a.CodiceGabbia = ad.CodiceGabbia
ORDER BY CodiceGabbia;
```

CodiceAnimale	Specie	Sesso	Eta	Dieta	CodiceGabbia	CodiceFiscale	Cognome
9	Leone	M	19	Carnivoro	3	PTTGM95M20C743Z	Pettenuzzo
19	Tigre	F	11	Carnivoro	4	BTTFNC94R01C743Q	Battistella
23	Panda Rosso	M	12	Carnivoro	5	NRDSFN96L17G693Z	Nordio
27	Mosasauro	M	1018	Carnivoro	6	MSRMRC92R06B563K	Masiero
24	Alligatore	F	17	Carnivoro	6	MSRMRC92R06B563K	Masiero

e) Un importante risultato scientifico di natura statistica riguarda l'età media degli animali con dieta differente. In un normale ecosistema, a causa delle stagioni e della pressante necessità di trovare cibo, solitamente gli animali erbivori tendono a vivere mediamente più a lungo di quelli carnivori. Ciò influisce ovviamente sull'equilibrio tra la specie, andando a regolare il numero degli esemplari presenti, le stagioni degli amori e moltissimi altri fattori importanti. Nonostante qui si parli di uno Zoo, si cerca comunque di preservare ogni Habitat nel migliore dei modi, con molta attenzione alle specie in esso introdotte. Un dato utile a prova della riuscita o meno di tale obiettivo è appunto il confronto tra l'età media degli animali con dieta opposta.

```
DROP VIEW IF EXISTS ETACarnivori;
CREATE VIEW ETACarnivori
AS
SELECT ETA(a.CodiceAnimale) as Eta, a.CodiceAnimale, c.Specie, c.Sesso, c.Dieta, a.CodiceGabbia
FROM Animale as a JOIN Cartella as c
ON a.CodiceAnimale=c.CodiceAnimale
WHERE c.Dieta='Carnivoro' AND a.CodiceAnimale<>27
ORDER BY Eta asc;
```

```
DROP VIEW IF EXISTS ETAErbivori;
CREATE VIEW ETAErbivori
AS
SELECT ETA(a.CodiceAnimale) as Eta, a.CodiceAnimale, c.Specie, c.Sesso, c.Dieta, a.CodiceGabbia
FROM Animale as a JOIN Cartella as c
ON a.CodiceAnimale=c.CodiceAnimale
WHERE c.Dieta='Erbivoro'
ORDER BY Eta asc;
```

```
SELECT CAST(AVG(EtaCarnivori.ETA) AS UNSIGNED) as EtaMediaCarnivori,
        CAST(AVG(EtaErbivori.ETA) AS UNSIGNED) as EtaMediaErbivori
FROM EtaCarnivori JOIN EtaErbivori;
```

EtaMediaCarnivori	EtaMediaErbivori
11	16

f) Una query che restituisca il Codice di tutti gli animali che non hanno mai ricevuto una Visita non riguardante un Vaccino, ossia quelli più in salute.

```
SELECT CodiceAnimale
FROM Animale
WHERE CodiceAnimale NOT IN (SELECT C.CodiceAnimale
                             FROM Visita AS V JOIN Cartella AS C
                             ON V.CodiceAnimale = C.CodiceAnimale
                             WHERE V.Motivazione<>'Vaccino')
ORDER BY CodiceAnimale ASC;
```

CodiceAnimale
2
4
6
8
9
10
11
12
13
15
16
17
19
21
23
24
27