

**Progetto Basi di Dati**

**Tropical Zoo**

Alessandro Lovo Amedeo Meggiolaro

ANNO ACCADEMICO 2017-2018

* 1. **Abstract**

“Tropical Zoo”, inaugurato nel 2007, è oggi un moderno Parco Zoologico che ha visto negli anni evolvere il proprio ruolo: da semplice esposizione di animali esotici ad attore fondamentale della loro conservazione. Gli obiettivi, tanto ambiziosi quanto stimolanti che il Parco si impegna a perseguire, riguardano la definizione di strategie che assicurino la sopravvivenza delle specie minacciate d’estinzione, la loro tutela e gestione in ambiente controllato, nonché focalizzare l’attenzione su temi quali la biologia di popolazione, il benessere dell’animale e la medicina veterinaria. Sotto l’aspetto educativo poi, la volontà è in primo luogo quella di educare le nuove generazioni al rispetto dell’ambiente, all’importanza della biodiversità e alla necessità di salvaguardare il mondo naturale. La ricerca è uno degli obiettivi fondamentali dei moderni giardini zoologici e quindi anche di Tropical Zoo. Ogni anno il personale del Parco infatti è a disposizione di molti studenti che scelgono di svolgere qui la loro tesi di laurea. La recente riorganizzazione delle aree all’interno del Parco, ha richiesto l’introduzione di sistemi informatici a supporto del personale. Tra questi è presente un database che ha lo scopo di migliorare la qualità e l’efficienza della gestione del Parco sotto vari aspetti, come: l’organizzazione del personale attraverso definizione della gerarchia e dei compiti in base alla tipologia di lavoratore; la catalogazione dei prodotti presenti nel negozio di souvenir; la registrazione delle vendite dei biglietti di ingresso; la disposizione degli animali all’interno delle gabbie nelle varie zone del Parco; lo storico dei controlli effettuati dai veterinari e altre informazioni che vengono visualizzate nel sito internet del Parco come ad esempio le tipologie e prezzi dei parcheggi e i contatti.

1. **Analisi dei requisiti**

Si vuole realizzare una base di dati per l’organizzazione di uno zoo. La sede è unica e possiede un recapito telefonico, un indirizzo e-mail, un orario di apertura al pubblico composto da giorno, ora apertura e ora chiusura, e un indirizzo formato da: città, via, numero civico, CAP.

Nel giardino zoologico lavorano diversi dipendenti, ad ognuno dei quali è assegnato un ruolo. Di ogni dipendente, identificato da un codice, si vogliono memorizzare: nome, cognome, salario e data di assunzione. La suddivisione in ruoli comporta diverse mansioni. I ruoli e le relative mansioni sono:

* Direttore: dirige il parco.
* Veterinario: si occupa della salute e il benessere degli animali, effettuando controlli periodici e programmati e intervenendo quando necessario.
* Cassiere: si occupa dell’emissione dei biglietti di ingresso al parco e della vendita di gadget e prodotti nel negozio di souvenir, posto sempre all’ingresso del parco.
* Keeper: addetto alla cura degli animali.
* Ricercatore: svolge attività di ricerca sugli animali in cattività e affianca gli studenti universitari nei propri lavori scientifici.
* Mascotte: intrattiene e diverte i più piccoli all’interno del parco.

Lo zoo è organizzato in aree, identificate da un codice progressivo e da un nome che rappresenta il continente o il paese di origine degli animali che contiene. In ogni area sono presenti alcune gabbie identificate univocamente da un codice, ognuna delle quali contiene una determinata specie di animali. Si vuole memorizzare inoltre il giorno di pulizia per ciascuna gabbia. Queste sono le informazioni che permettono di identificare ogni esemplare presente nel parco:

* codice univoco
* nome comune
* nome scientifico (specie)
* famiglia
* ordine
* classe
* sesso
* età
* data di nascita
* data di arrivo nello zoo
* codice gabbia

Sempre per quanto riguarda gli animali è necessario salvare nella base di dati i controlli medici effettuati dai veterinari ai vari esemplari. Per poter chiarire i risultati di ciascun controllo si rappresentano le seguenti informazioni:

* codice animale
* data controllo
* peso animale
* malattia (0,1)
* codice veterinario

Oltre alla gestione del personale e degli animali nelle gabbie e nelle varie aree, la base di dati ha lo scopo di catalogare i vari gadget in vendita nel negozio di souvenir posto all’interno del giardino zoologico e controllarne la disponibilità. Il singolo prodotto ha un codice identificativo, un prezzo, appartiene ad una determinata tipologia di prodotto (poster, maglietta, libro, penna, …) e in ogni momento è possibile conoscere la quantità della scorta rimanente.

Viene infine tenuta traccia dei biglietti venduti ai visitatori. Per fare questo ogni biglietto possiede un codice univoco e viene inoltre memorizzata la categoria del ticket emesso, costituita dal prezzo corrispondente e identificata da un nome.

**Glossario dei termini**

|  |  |  |
| --- | --- | --- |
| **Termine** | **Descrizione** | **Collegamento** |
| Zoo | Il Parco zoologico “Tropical Zoo” | Animale, Area, Negozio\_souvenir, Dipendente |
| Visitatore | Persona che visita il parco acquistando un biglietto di ingresso | Zoo |
| Animale | Un animale ospitato all’interno del parco zoologico “Tropical Zoo” | Gabbia, Controllo\_medico |
| Gabbia | Gabbia contenuta in un’area dello zoo che contiene una determinata specie di animali | Animale, Area |
| Area | Area dello zoo contenente un certo numero di gabbie | Gabbia, Zoo |
| Dipendente | Si suddivide in: direttore, veterinario, cassiere, keeper, ricercatore, mascotte | Controllo\_medico, Zoo |
| Controllo\_medico | Controllo effettuato da uno o più veterinari ad un animale | Veterinario, Animale |
| Biglietto | Ticket acquistato dai visitatori per accedere al parco | Visitatore |
| Negozio\_souvenir | Negozio situato all’interno del parco dove i visitatori possono acquistare vari gadget | Zoo, Merce |
| Merce | Gadget e prodotti in vendita nel negozio di souvenir | Negozio\_souvenir |

**Strutturazione dei requisiti**

|  |
| --- |
| **FRASI RELATIVE A *ZOO*** |
| La sede è unica e possiede un recapito telefonico, un indirizzo e-mail, un orario di apertura al pubblico composto da giorno, ora apertura e ora chiusura, e un indirizzo formato da: città, via, numero civico, CAP. Nel giardino zoologico lavorano diversi dipendenti… è organizzato in aree… identificare ogni esemplare presente nel parco… negozio di souvenir posto all’interno del giardino zoologico… |
| **FRASI RELATIVE AD *ANIMALE*** |
| Lo zoo è organizzato in aree… in ogni area sono presenti alcune gabbie… ognuna delle quali contiene una determinata specie di animali. Queste sono le informazioni che permettono di identificare ogni esemplare presente nel parco:   * codice univoco * nome comune * nome scientifico (specie) * famiglia * ordine * classe * sesso * età * data di nascita * data di arrivo nello zoo * codice gabbia   Sempre per quanto riguarda gli animali è necessario salvare i controlli medici effettuati dai veterinari ai vari esemplari. |
| **FRASI RELATIVE A *DIPENDENTE*** |
| Nel giardino zoologico lavorano diversi dipendenti, ad ognuno dei quali è assegnato un ruolo. Di ogni dipendente, identificato da un codice, si vogliono memorizzare: nome, cognome, salario e data di assunzione. La suddivisione in ruoli comporta diverse mansioni. I ruoli e le relative mansioni sono:   * Direttore: dirige il parco. * Veterinario: si occupa della salute e il benessere degli animali, effettuando controlli periodici e programmati e intervenendo quando necessario. * Cassiere: si occupa dell’emissione dei biglietti di ingresso al parco e della vendita di gadget e prodotti nel negozio di souvenir, posto sempre all’ingresso del parco. * Keeper: addetto alla cura degli animali. * Ricercatore: svolge attività di ricerca sugli animali in cattività e affianca gli studenti universitari nei propri lavori scientifici. * Mascotte: intrattiene e diverte i più piccoli all’interno del parco. |
| **FRASI RELATIVE A *CONTROLLO\_MEDICO*** |
| …è necessario salvare nella base di dati i controlli medici effettuati dai veterinari ai vari esemplari. Per poter chiarire i risultati di ciascun controllo si rappresentano le seguenti informazioni:   * codice animale * data controllo * peso animale * malattia (0,1) * codice veterinario |
| **FRASI RELATIVE AD *AREA*** |
| Lo zoo è organizzato in aree, identificate da un codice progressivo e da un nome che rappresenta il continente o il paese di origine degli animali che contiene. In ogni area sono presenti alcune gabbie… |
| **FRASI RELATIVE A *GABBIA*** |
| In ogni area sono presenti alcune gabbie identificate univocamente da un codice, ognuna delle quali contiene una determinata specie di animali. Si vuole memorizzare inoltre il giorno di pulizia per ciascuna gabbia. |
| **FRASI RELATIVE A VISITATORE** |
| Viene tenuta traccia dei biglietti venduti ai visitatori. |
| **FRASI RELATIVE A *BIGLIETTO*** |
| Ogni biglietto possiede un codice univoco e viene inoltre memorizzata la categoria del ticket emesso, costituita dal prezzo relativo e identificata da un nome. |
| **FRASI RELATIVE A *NEGOZIO\_SOUVENIR*** |
| …la base di dati ha lo scopo di catalogare i vari gadget in vendita nel negozio di souvenir posto all’interno del giardino zoologico e controllarne la disponibilità. |
| **FRASI RELATIVE A *MERCE*** |
| Il singolo prodotto ha un codice identificativo, un prezzo, appartiene ad una determinata tipologia di prodotto (poster, maglietta, libro, penna, …) e in ogni momento è possibile conoscere la quantità della scorta rimanente. |

1. **Progettazione logica**

**Lista delle entità**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zoo** | | | |
| Telefono | INTEGER | Numero di telefono dello zoo | |
| Email | VARCHAR | Indirizzo e-mail dello zoo | |
| Indirizzo | Attributo composto: Citta(VARCHAR), Via(VARCHAR), CAP (INTEGER) | | |
| Orario | Attributo composto: Giorno (VARCHAR), Apertura (TIME), Chiusura (TIME) | | |
| **Animale** | | | |
| Id\_animale | VARCHAR | Identifica univocamente un animale | Chiave |
| Specie | VARCHAR | Specie dell’esemplare | |
| Nome\_comune | VARCHAR | Nome comune dell’esemplare | |
| Classe | VARCHAR | Nome della classe a cui appartiene l’esemplare | |
| Ordine | VARCHAR | Nome dell’ordine a cui appartiene l’esemplare | |
| Famiglia | VARCHAR | Nome della famiglia a cui appartiene l’esemplare | |
| Sesso | VARCHAR | Sesso dell’esemplare (M/F) | |
| Eta | INTEGER | Età dell’esemplare | |
| Data\_arrivo | DATE | Data di arrivo allo zoo dell’esemplare (se non è nato allo zoo) | |
| Data\_nascita | DATE | Data di nascita dell’esemplare (se nato allo zoo o se il dato è disponibile) | |
| **Dipendente** | | | |
| Id\_dipendente | VARCHAR | Identifica univocamente un dipendente | Chiave |
| Nome | VARCHAR | Nome del dipendente | |
| Cognome | VARCHAR | Cognome del dipendente | |
| Data\_assunzione | DATE | Data di inizio del lavoro presso il parco zoologico | |
| Salario | DECIMAL | Indica la retribuzione corrisposta al dipendente | |
| **Direttore** | | | |
| Nessun attributo | | | |
| **Veterinario** | | | |
| Nessun attributo | | | |
| **Cassiere** | | | |
| Nessun attributo | | | |
| **Keeper** | | | |
| Nessun attributo | | | |
| **Ricercatore** | | | |
| Nessun attributo | | | |
| **Mascotte** | | | |
| Nessun attributo | | | |
| **Controllo\_medico** | | | |
| Id\_animale | VARCHAR | Identifica univocamente l’esemplare sottoposto a controllo medico | Chiave |
| Giorno | VARCHAR | Data in cui il veterinario ha eseguito il controllo |
| Id\_dipendente | VARCHAR | Identifica univocamente il veterinario che ha eseguito il controllo |
| Peso | DECIMAL | Peso in chilogrammi dell’animale al momento del controllo | |
| Malattia | INTEGER | Indica se sono state rilevate criticità nello stato di salute dell’animale durante il controllo (valore 1 se in stato di malattia, 0 altrimenti) | |
| **Area** | | | |
| Id\_area | INTEGER | Identifica univocamente un’area del giardino zoologico | Chiave |
| Nome | VARCHAR | Nome assegnato ad ogni area | |
| N\_gabbie | INTEGER | Numero di gabbie presenti in ogni area | |
| **Gabbia** | | | |
| Id\_gabbia | INTEGER | Identifica univocamente una gabbia | Chiave |
| N\_animali | INTEGER | Numero di animali presenti all’interno di una gabbia | |
| Giorno\_pulizia | VARCHAR | Giorno della settimana in cui una gabbia viene pulita | |
| Capienza | INTEGER | Numero massimo di animali che può contenere la gabbia | |
| **Biglietto** | | | |
| Id\_biglietto | VARCHAR | Identifica in modo univoco un ticket emesso per l’ingresso al parco | Chiave |
| Tipologia\_biglietto | Attributo composto: Nome (PRIMARY KEY, VARCHAR), Prezzo (DECIMAL) | | |
| Giorno | Data di emissione del biglietto | | |
| **Merce** | | | |
| Id\_merce | VARCHAR | Identifica univocamente una tipologia di articolo in vendita presso il negozio souvenir dello zoo | Chiave |
| Prezzo | DECIMAL | Prezzo applicato all’oggetto in vendita | |
| Tipo | VARCHAR | Tipologia di prodotto (poster, maglietta, libro, penna, …) | |
| **Negozio\_souvenir** | | | |
| Id\_merce | VARCHAR | Identifica un articolo presente nel negozio | Chiave |
| Quantita | INTEGER | Quantità rimanente di un certo prodotto nel negozio | |

**Approfondimenti**

**Generalizzazioni**

* Dipendente è generalizzazione completa ed esclusiva delle entità: Direttore, Veterinario, Cassiere, Keeper, Ricercatore, Mascotte.

L’attributo composto Indirizzo sull’entità Zoo si compone di tre concetti: Città, CAP e Via che

dovrà contenere anche il numero civico.

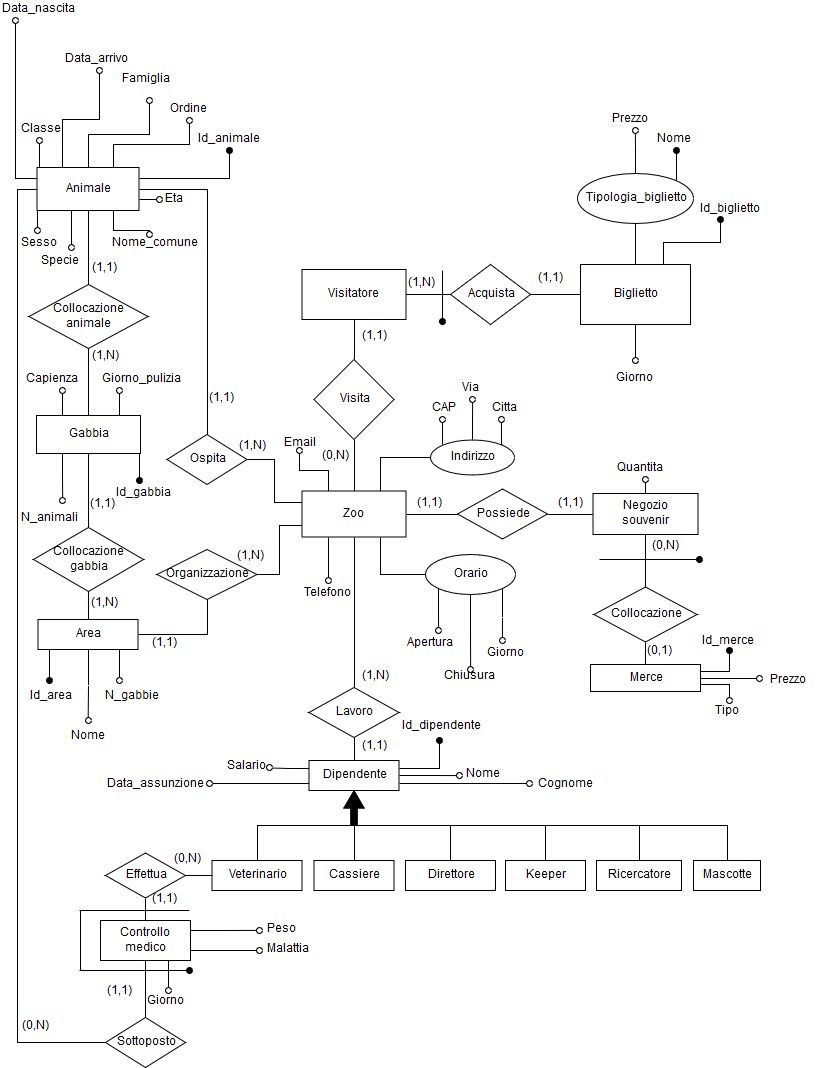
L’attributo composto Orario sull’entità Zoo si compone di tre concetti: Giorno, Apertura (orario), Chiusura (orario).

L’attributo composto Categoria sull’entità Biglietto si compone di due concetti importanti: Nome e Prezzo.

**Analisi delle relazioni e delle cardinalità**

* **Zoo – Visitatore: Visita**
  + Lo zoo è visitato da uno o più visitatori (0, N)
  + Un visitatore visita un solo zoo (1, 1)
* **Zoo – Animale: Ospita**
  + Nello zoo sono ospitati uno o più animali (1, N)
  + Un animale è ospitato in uno zoo (1, 1)
* **Zoo – Area: Organizzazione**
* Lo zoo è organizzato in una o più aree (1, N)
* Un’area appartiene ad un unico zoo (1, 1)
* **Zoo – Dipendente: Lavoro**
* Nello zoo devono lavorare uno o più dipendenti (1, N)
* Un dipendente lavora in un unico zoo (1, 1)
* **Zoo – Negozio\_souvenir: Possiede**
* Lo zoo possiede e gestisce un negozio souvenir (1, 1)
* Il negozio souvenir è gestito solo da uno zoo (1, 1)
* **Negozio\_souvenir – Merce: Collocazione**
  + Nel negozio di souvenir sono presenti uno o più prodotti in vendita (0, N)
  + Un prodotto è collocato in un solo negozio souvenir (0, 1)
* **Veterinario – Controllo\_medico: Effettua**
* Un veterinario effettua uno o più controlli sugli animali (0, N)
* Un controllo è effettuato da un solo veterinario (1, 1)
* **Animale – Controllo\_medico: Sottoposto**
  + Un animale è sottoposto a uno o più controlli (0, N)
  + Un controllo riguarda un solo animale (1, 1)
* **Animale – Gabbia: Collocazione animale**
* Un animale vive in una gabbia (1, 1)
* In una gabbia possono esserci uno o più animali (1, N)
* **Gabbia – Area: Collocazione gabbia**
  + Una gabbia è collocata in una sola area (1, 1)
  + In un’area sono collocate una o più gabbie (1, N)

**Schema ER**



1. **Progettazione logica**

**Analisi delle ridondanze**

Nel database sono state riscontrate due ridondanze:

1. **Attributo N\_animali in Gabbia**

Una ridondanza si trova nell’ entità Gabbia, nella quale il valore dell’attributo N\_animali​ si può calcolare visitando la relazione ​Collocazione animale​ (la relazione presente tra l’entità Animale e Gabbia).

1. **Attributo N\_gabbie in Area**

Un’ altra ridondanza si trova nell’entità Area, nella quale il valore dell’attributo N\_gabbie si può calcolare visitando la relazione ​Collocazione gabbia​ (la relazione presente tra l’entità Area e Gabbia).

**Tabella dei volumi**

|  |  |  |
| --- | --- | --- |
| **Concetto** | **Tipo** | **Volume** |
| Animale | E | 40 |
| Collocazione animale | R | 40 |
| Gabbia | E | 30 |
| Collocazione gabbia | R | 30 |
| Area | E | 9 |

**Tabelle delle operazioni**

1. **Attributo N\_animali in gabbia**

|  |  |  |
| --- | --- | --- |
| **Operazione** | **Tipo** | **Frequenza** |
| 1. Verifica del numero di posti rimasti liberi in una gabbia | I | 60 volte al giorno\* |

\*Operazione effettuata 2 volte al giorno per ciascuna gabbia.

|  |  |  |  |
| --- | --- | --- | --- |
| **OP. 1 – CON RIDONDANZA** | | | |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Gabbia | Entità | 2 | L |
| Poiché l’attributo N\_animali è presente sarà sufficiente reperire questo dato e l’attributo Capienza direttamente dall’entità gabbia per calcolare il risultato richiesto. | | | |
| Totale accessi: 2\*60 = 120 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **OP. 1 – SENZA RIDONDANZA** | | | |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Animale | Entità | 1 | L |
| Collocazione animale | Relazione | 1 | L |
| Gabbia | Entità | 1 | L |
| Questa operazione implica il reperimento degli animali che vivono nella gabbia che si vuole esaminare. In base alla tabella dei volumi, consideriamo di fare in media un accesso all’entità Animale per ogni esecuzione dell’operazione, e quindi anche alla relazione Collocazione animale. È necessario accedere poi una volta all’entità Gabbia per ottenere l’attributo Capacità per calcolare il risultato richiesto. | | | |
| Totale accessi: 3\*60 = 180 | | | |

In conclusione, dopo aver esaminato il totale degli accessi per l’operazione 1 con e senza ridondanza, si è deciso di mantenere l’attributo N\_animali in Gabbia. Infatti con la ridondanza si ha un risparmio di 60 accessi giornalieri.

1. **Attributo N\_gabbie in Area**

|  |  |  |
| --- | --- | --- |
| **Operazione** | **Tipo** | **Frequenza** |
| 2. Verifica il numero totale di gabbie collocate in un’area | I | 18 volte al giorno\* |

\*Operazione effettuata 2 volte al giorno per ciascuna area.

|  |  |  |  |
| --- | --- | --- | --- |
| **OP. 2 – CON RIDONDANZA** | | | |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Area | Entità | 1 | L |
| Poiché l’attributo N\_gabbie è presente sarà sufficiente reperire individualmente questo dato. | | | |
| Totale accessi: 1\*18 = 18 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **OP. 2 – SENZA RIDONDANZA** | | | |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Gabbia | Entità | 3 | L |
| Collocazione gabbia | Relazione | 3 | L |
| Area | Entità | 1 | L |
| Questa operazione implica il reperimento delle gabbie che sono collocate nell’area che si vuole esaminare. In base alla tabella dei volumi, consideriamo di fare in media 3 accessi all’entità Gabbia per ogni esecuzione dell’operazione, e quindi anche alla relazione Collocazione gabbia. | | | |
| Totale accessi: 7\*18 = 126 | | | |

In conclusione, dopo aver esaminato il totale degli accessi per l’operazione 2 con e senza ridondanza, si è deciso di mantenere l’attributo N\_gabbie in Area. Infatti, con la ridondanza si ha un risparmio di 108 accessi giornalieri.

**Eliminazione delle generalizzazioni**

* **Dipendente**

Questa generalizzazione presenta diverse entità figlie. Si procede ad una valutazione sotto vari aspetti al fine di determinare il tipo di implementazione delle generalizzazioni nello schema logico.

ATTRIBUTI: L’entità padre presenta degli attributi che sono comuni a tutte le entità figlie, e a loro volta, le entità figlie non presentano attributi specifici che le distinguono tra di loro e dall’entità padre.

OPERAZIONI: Sono presenti sia operazioni che non fanno distinzione tra entità padre ed entità figlie, sia operazioni che fanno riferimento in modo distinto a entità figlie ed entità padre.

Essendo una generalizzazione totale ed esclusiva si è deciso di accorpare le entità figlie in un nuovo attributo Impiego all’interno dell’entità Dipendente per identificare la mansione del dipendente. Il motivo di questa scelta è il fatto che se si fosse deciso di sostituire la generalizzazione con l’accorpamento dell’entità genitore nelle entità figlie si sarebbero dovute creare 6 entità (una per ogni entità figlia) e ripetere i 5 attributi dell’entità padre in ogni entità figlia con un conseguente spreco di memoria per attributi che non fanno distinzione tra entità figlie.

**Introduzione entità Orario**

Per rappresentare le informazioni riguardanti l’orario di apertura dello zoo, è stata introdotta l’entità Orario. Questa entità contiene i seguenti attributi: Giorno, Apertura, Chiusura. L’entità Orario è collegata all’entità Zoo tramite la relazione Attività.

**Introduzione entità Tipologia\_biglietto**

Si è scelto di introdurre una nuova entità Tipologia\_biglietto per identificare le diverse tariffe in base all’età dei visitatori. Questa entità contiene i seguenti attributi: Nome, Prezzo, Eta\_min, Eta\_max. L’entità è collegata all’entità Biglietto tramite la relazione Tipo.

**Eliminazione entità Visitatore**

Per come è schematizzata nel modello concettuale, l’entità Visitatore non è utile in quanto un visitatore viene identificato solamente dall’attributo Id\_biglietto dell’entità Biglietto e per questo motivo si è deciso di eliminarla. È stata quindi aggiunta la relazione Ingresso tra le entità Zoo e Biglietto.

**Vincoli di integrità referenziale**

**Biglietto – Tipologia\_biglietto**: vincolo di integrità referenziale tra Tipologia in Biglietto e Nome in Tipologia\_biglietto.

**Gabbia – Area**: vincolo di integrità referenziale tra Id\_area in Gabbia e Id\_area in Area.

**Animale – Gabbia**: vincolo di integrità referenziale tra Id\_gabbia in Animale e Id\_gabbia in Gabbia.

**Controllo\_medico – Animale**: vincolo di integrità referenziale tra Id\_animale in Controllo\_medico e Id\_animale in Animale.

**Controllo\_medico – Dipendenti**: vincolo di integrità referenziale tra Id\_veterinario in Controllo\_medico e Id\_dipendente in Dipendenti.

**Negozio\_souvenir – Merce**: vincolo di integrità referenziale tra Id\_merce in Negozio\_souvenir e Id\_merce in Merce.

**Traduzione verso il modello relazionale**

**ZOO** (Telefono, Email, Citta, Via, CAP)

**ORARIO** (Giorno, Apertura, Chiusura)

**TIPOLOGIA\_BIGLIETTO**: (Nome, Prezzo, Eta\_min, Eta\_max)

**BIGLIETTO** (Id\_biglietto, Tipologia, Giorno)

**DIPENDENTE** (Id\_dipendente, Impiego, Nome, Cognome, Data\_assunzione, Salario)

**AREA** (Id\_area, Nome)

**GABBIA** (Id\_gabbia, Id\_area, Giorno\_pulizia, Capienza)

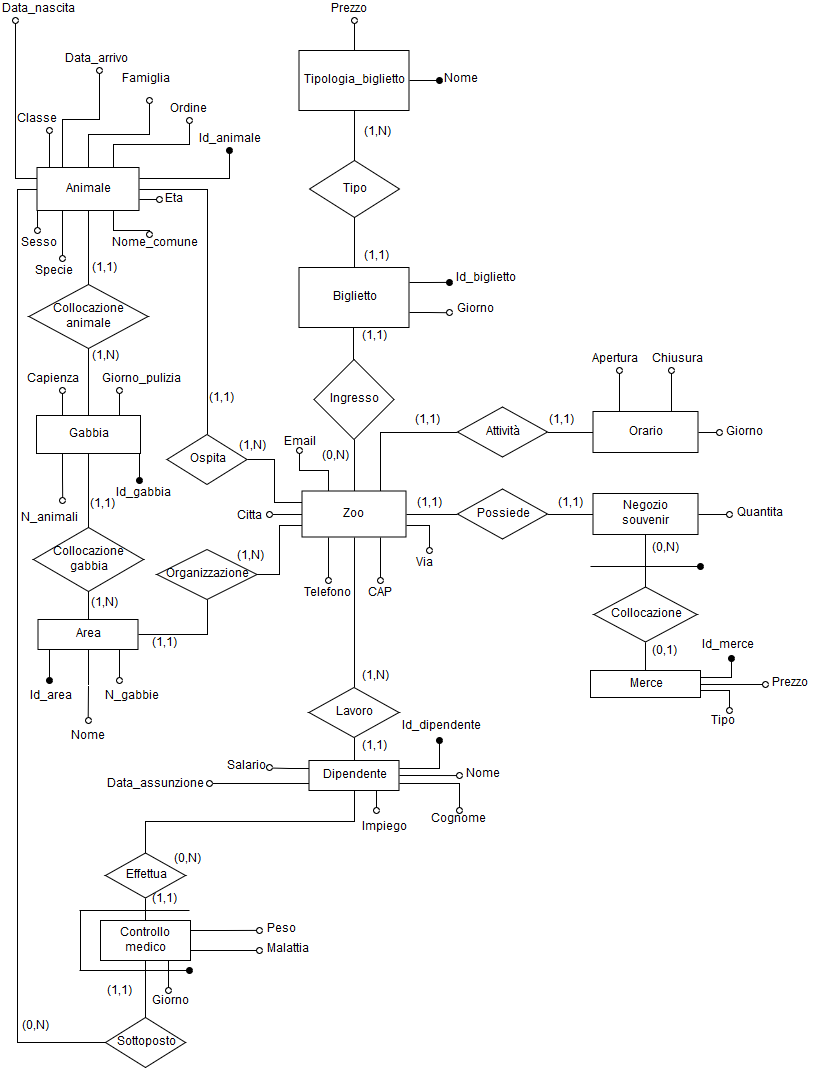
**ANIMALE** (Id\_animale, Id\_gabbia, Nome\_comune, Specie, Classe, Ordine, Famiglia, Sesso, Eta, Data\_arrivo, Data\_nascita)

**CONTROLLO\_MEDICO** (Id\_animale, Id\_veterinario, Giorno, Peso, Malattia)

**MERCE** (Id\_merce, Prezzo, Tipo)

**NEGOZIO\_SOUVENIR** (Id\_merce, Quantita)

**Schema ER ristrutturato**



1. **Query, Procedure, trigger e funzioni**

**Funzioni**

* 1. **Si vuole creare una funzione che restituisca il numero di controlli effettuati in un certo periodo di tempo da un veterinario. Essa prende in input l’id dipendente del veterinario e le date di inizio e fine del periodo da analizzare.**

DELIMITER $$

CREATE FUNCTION ControlliEseguiti(inizio DATE, fine DATE, veterinario VARCHAR(20)) RETURNS int(10) unsigned

BEGIN

DECLARE n\_controlli INTEGER;

SELECT COUNT(\*) INTO n\_controlli FROM Controllo\_medico WHERE Id\_veterinario = veterinario AND Giorno BETWEEN inizio AND fine;

RETURN n\_controlli;

END$$

DELIMITER ;

* 1. **Si vuole creare una funzione che restituisca lo stato di salute attuale di un animale in base all’ultimo controllo medico effettuato.**

DELIMITER $$

CREATE FUNCTION Malattia(animale VARCHAR(10)) RETURNS int(11)

BEGIN

DECLARE malattia INTEGER;

SELECT c1.Malattia INTO malattia FROM Controllo\_medico AS c1 JOIN(

SELECT MAX(Giorno) AS MaxCheckData, Id\_animale

FROM Controllo\_medico

WHERE Id\_animale = animale

) AS c2

ON c1.Id\_animale = c2.Id\_animale

AND c1.Giorno = c2.MaxCheckData;

RETURN malattia;

END$$

DELIMITER ;

**Query e procedure**

* + 1. **Si vogliono elencare gli animali presenti nell’area con Id\_area = 4.**

SELECT Animale.Id\_animale, Animale.Nome\_comune, Gabbia.Id\_gabbia, Area.Nome AS Nome\_Area FROM Animale JOIN Gabbia ON Animale.Id\_gabbia = Gabbia.Id\_gabbia JOIN Area ON Gabbia.Id\_area = Area.Id\_area WHERE Gabbia.Id\_area = 4;



* + 1. **Si vogliono eliminare tutti i biglietti e i controlli medici effettuati in un determinato anno.**

DELIMITER $$

CREATE PROCEDURE DeleteLog(anno INT(4) UNSIGNED)

BEGIN

DELETE FROM Biglietto WHERE YEAR(Giorno) = anno;

DELETE FROM Controllo\_medico WHERE YEAR(Giorno) = anno;

END$$

DELIMITER ;

**CALL DeleteLog(2017)**

* 1. **Si vogliono visualizzare i prodotti presenti nel negozio di souvenir in quantità minore di quella specificata.**

DELIMITER $$

CREATE PROCEDURE Scorte(scorta INT(3) UNSIGNED)

SELECT Negozio\_souvenir.Id\_merce, Prezzo, Tipo, Quantita as Rimanenti FROM Negozio\_souvenir, Merce WHERE Negozio\_souvenir.Id\_merce = Merce.Id\_merce AND Quantita < scorta;

END$$

DELIMITER ;

**CALL Scorte(30)**



* 1. **Si vogliono visualizzare gli animali attualmente malati.**

CREATE VIEW AnimaliMalati AS SELECT Id\_animale, Giorno FROM Controllo\_medico WHERE Malattia(Controllo\_medico.Id\_animale) = 1;

SELECT Animale.Id\_animale, Nome\_comune AS 'Nome comune', Id\_gabbia AS 'Gabbia', Giorno AS 'Ultimo Controllo' FROM AnimaliMalati JOIN Animale WHERE AnimaliMalati.Id\_animale = Animale.Id\_animale;



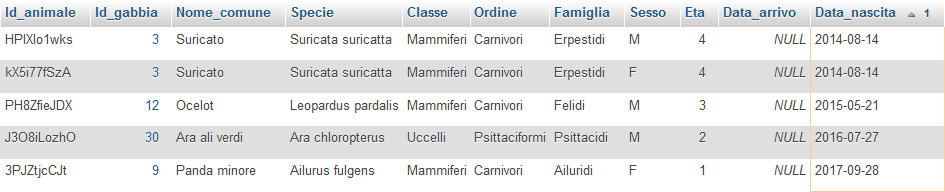
* 1. **Si vogliono elencare le gabbie con numero di animali ospitati pari alla capienza massima della gabbia.**

SELECT Id\_gabbia, Id\_area FROM Gabbia WHERE Gabbia.N\_animali=Gabbia.Capienza;



1. **Si vogliono elencare gli animali nati nello zoo ordinati per età decrescente.**

SELECT \* FROM Animale WHERE Data\_nascita IS NOT NULL ORDER BY Animale.Data\_nascita;



**Trigger**

* 1. **Trigger che blocca l’inserimento di nuovi biglietti con campo Id\_biglietto già esistente nel database.**

DELIMITER //

CREATE TRIGGER BloccoDuplicatoBiglietto

BEFORE INSERT ON Biglietto

FOR EACH ROW

BEGIN

DECLARE msg VARCHAR(255);

IF (EXISTS(SELECT 1 FROM Biglietto WHERE Id\_biglietto = NEW.Id\_biglietto)) THEN

SET msg = "Errore: id biglietto già esistente. Generare un altro id.";

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

END IF;

END //

DELIMITER ;

#1644 - Errore: id biglietto già esistente. Generare un altro id.

* 1. **Trigger che effettua controlli prima dell’inserimento di nuovi animali nel database.**

DELIMITER //

CREATE TRIGGER ErratoInserimentoAnimale

BEFORE INSERT ON Animale

FOR EACH ROW

BEGIN

DECLARE msg VARCHAR(255);

DECLARE num INTEGER;

DECLARE capacita INTEGER;

DECLARE specie\_animale VARCHAR(50);

SELECT Specie INTO specie\_animale FROM Animale WHERE Animale.Id\_gabbia = NEW.Id\_gabbia LIMIT 1;

SELECT N\_animali INTO num FROM Gabbia WHERE Gabbia.Id\_gabbia = NEW.Id\_gabbia;

SELECT Capienza INTO capacita FROM Gabbia WHERE Gabbia.Id\_gabbia = NEW.Id\_gabbia;

IF (EXISTS(SELECT 1 FROM Animale WHERE Id\_animale = NEW.Id\_animale)) THEN

SET msg = "Errore: id animale già esistente. Generare un altro id.";

END IF;

IF (NEW.Data\_arrivo IS NOT NULL AND NEW.Data\_nascita IS NOT NULL) THEN

SET msg = "Errore: inserire solamente data di nascita o data di arrivo nello zoo.";

END IF;

IF capacita = num THEN

SET msg = "Errore: non c'è posto nella gabbia per il nuovo animale. Selezionare un'altra gabbia.";

END IF;

IF (specie\_animale != NEW.Specie) THEN

SET msg = "Errore: Nella gabbia sono presenti animali di un'altra specie. Selezionare un'altra gabbia.";

END IF;

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

END //

DELIMITER ;

#1644 - Errore: id animale già esistente. Generare un altro id.

#1644 - Errore: inserire solamente data di nascita o data di arrivo nello zoo.

#1644 - Errore: non c'è posto nella gabbia per il nuovo animale. Selezionare un'altra gabbia.

#1644 - Errore: Nella gabbia sono presenti animali di un'altra specie. Selezionare un'altra gabbia.

* 1. **Trigger che aggiorna il numero di animali in una gabbia dopo l’inserimento di un nuovo esemplare in quella gabbia.**

DELIMITER //

CREATE TRIGGER UpdateNumeroAnimali

AFTER INSERT ON Animale

FOR EACH ROW

BEGIN

DECLARE num INTEGER;

UPDATE Gabbia SET N\_animali = N\_animali + 1 WHERE Gabbia.Id\_gabbia = NEW.Id\_gabbia;

END //

DELIMITER ;