

Progetto Basi Di Dati AA 2018-2019

ZAO FOX VILLAGE

1171044 – MARCEL JUNIOR WANDJI (9 CFU)
1169656- ANDREI NICOLAE SPIRIDONESCU
(9 CFU)

Abstract

Lo Zao Fox Village, localmente conosciuto come "Kitsune Mura" è un parco zoologico situato in Giappone. Il parco è interamente dedicato alle volpi, avente piu' di 200 esemplari suddivisi in diverse specie. Le volpi sono cresciute in cattività e sono abituate alla presenza umana tanto da essere considerate semi-domestiche. Il parco è suddiviso in settori, ognuno interamente dedicato ad una specie dove le volpi possono muoversi in libertà ed interagire fra di loro. Il parco è aperto al pubblico che può accerderci prenotando una visita al costo di 1000 yen (circa 8 euro). Data la domestichezza delle volpi al contatto umano, all'interno del villaggio i visitatori potranno entrare direttamente in contatto con gli esemplari, avendo la possibilità di accarezzare o tenere in braccio i cuccioli , di passeggiare e visitare i vari settori del parco. Inoltre si ha la possibilità di dare da mangiare agli esemplari. Lo Zao fox village rimane aperto per tutta la durata di tutto l'anno con l'eccezione di alcuni giorni festivi segnati sul calendario presente sul sito ufficiale del parco. Ulteriori informazioni sono disponibili sul sito web aperto al pubblico mentre per l'amministrazione interna del parco è gestita da una base di dati elettronica non piu' cartacea in modo da migliorare l'amministrazione.

Descrizione dei requisiti

Si vuole realizzare una base di dati che contenga e gestisca le informazioni relative al parco zoologico chiamato "Zao fox village". Lo Zao fox village è interamente dedicato alle volpi, avente piu di 200 esemplari appartenenti a diverse specie e aventi un id che le identifica e una data di nascita. Ogni specie ha un id e un nome ed è localizzata in un settore avente un nome. Gli impiegati che ci lavorano sono : medico, guardia e hostess. Ogni impiegato ha un nome, cognome, data di nascita, data di assunzione e un id. I medici si occupano delle visite mediche delle volpi. Per ogni visita effettuata si memorizza la data. Ogni settore è sorvegliato da guardie. Ogni risorsa ha un id, un nome ed un prezzo , e viene fornita da fornitori(aventi nomi e id) e può essere di tipo alimentare o medica. Le risorse alimentari hanno una data di scadenza e sono usate per la nutrizione delle specie e quelle mediche per le visite mediche. Delle prenotazioni affettuate dai clienti, si vuole memorizzare la data e il numero della prenotazione. Ogni cliente ha un nome ,cognome e una email che lo identifica. A ogni prenotazione viene assegnata un hostess che accompagna il cliente durante la visita turistica.

Progettazione Concettuale

Entità e Attributi

Impiegato

id : int

nome : string

cognome : string

dataNascita: date

dataAssunzione : date

Medico

Guardia

Fornitore

id : int

nome : string

Settore

nome : char

Hostess

Visita

dataVisita : date

Specie

id : int

nome : string

Alimentare

id : int

nome : string

prezzo : int

dataScadenza : date

Risorsa

id : int

nome : string

prezzo : int

Prenotazione

numero : string

data : date

Medica

Cliente

email : string

nome : string

cognome : string

Volpe

id : string

dataNascita : date

Relazioni

Medico-Visita :effettua

-un medico effettua zero o più visite

-una visita è effettuata da un un medico

Guardia-Settore : occupa

-una guardia sorveglia zero o un settore

-un settore è sorvegliato da almeno una guardia

Hostess-Prenotazione : gestisce

-un hostess può essere associato a nessuna o più prenotazioni

-una prenotazione ha un hostess

Prenotazione-Cliente : prenota

- un cliente ha almeno una prenotazione
- una prenotazione è fatta da un cliente

Fornitore-Risorsa : fornisce

- un fornitore fornisce almeno una risorsa
- una risorsa è fornita da un fornitore

Medica-Visita : usa

- una risorsa medica è usata in nessuna o più visite
- un visita usa nessuna o una risorsa

Alimentare-Specie : mangia

- un alimento è mangiato da almeno una specie
- una specie mangia un alimento

Visita-Volpe : su

- una visita è effettuata una volta su una volpe in una data
- una volpe è visitata almeno una volta

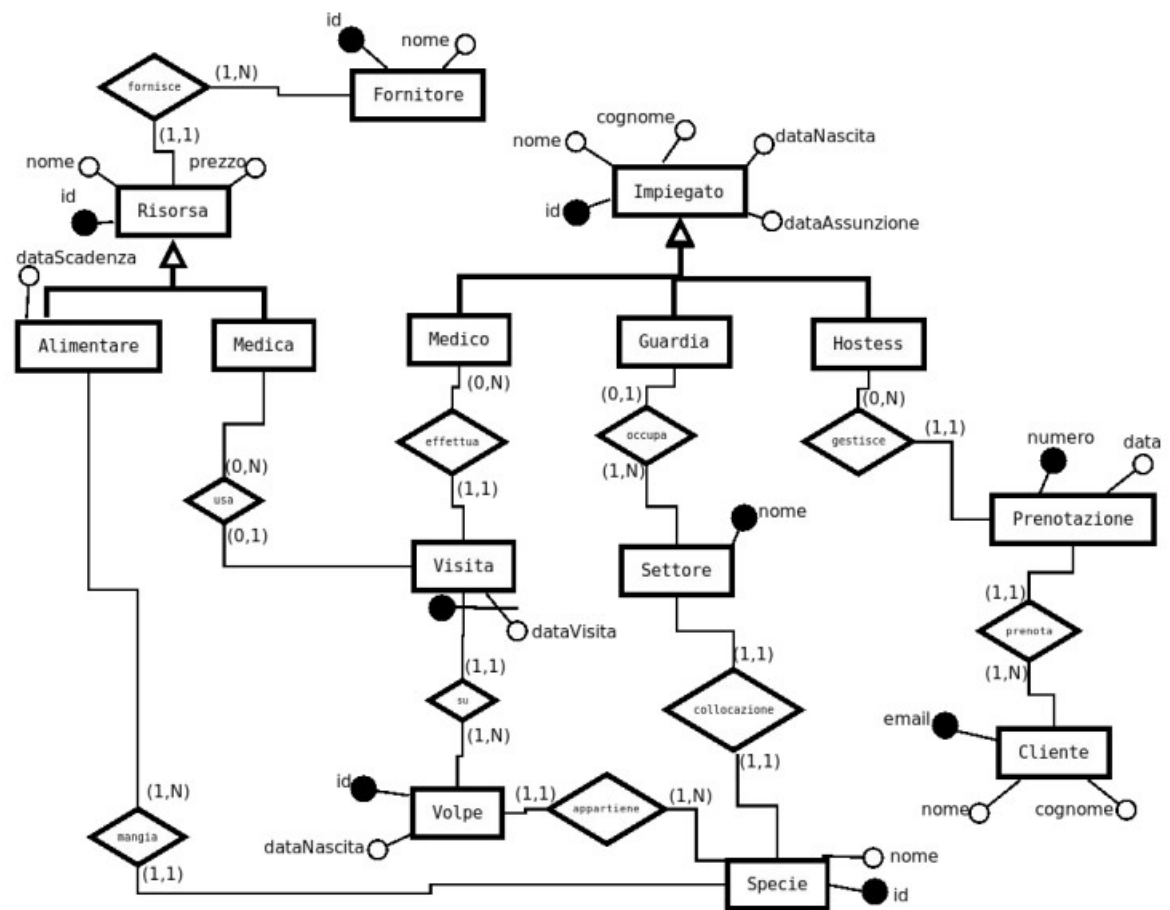
Volpe-Specie : appartiene

- una volpe appartiene a una sola specie
- c'è almeno una volpe in una specie

Settore-Specie : collocazione

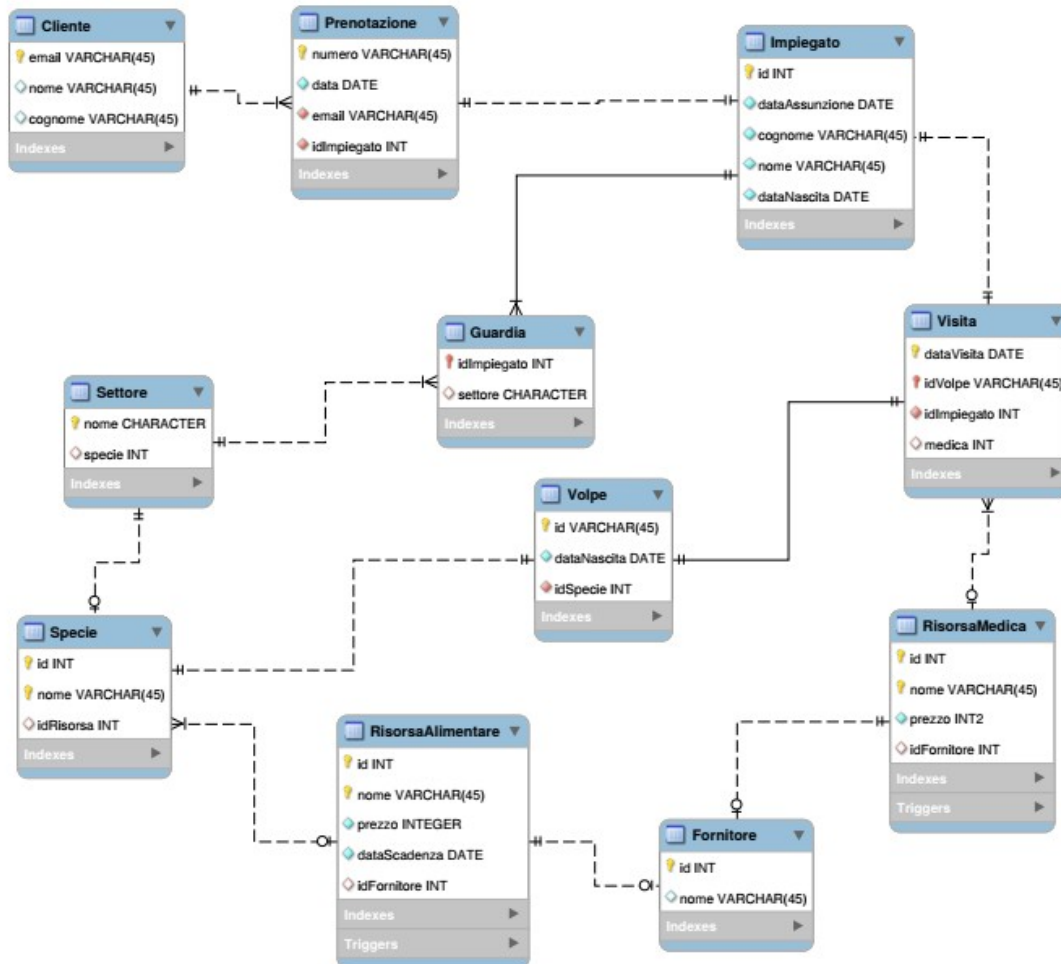
- un settore contiene una specie
- una specie è in un settore

DIAGRAMMA ER



SCHEMA LOGICO

Thu Feb 7 20:06:55 2019, New Model - EER Diagram (part 1 of 2)



SCHEMA RELAZIONALE

Cliente(email, nome , cognome)

Prenotazione(numero, data, email, idImpiegato)

Impiegato(id, dataAssunzione,cognome, nome, dataNascita)

Visita(dataVisita,idVolpe, idImpiegato, medica)

Guardia(idImpiegato, settore)

Settore(nome,specie)

Volpe(id, dataNascita, idSpecie)

Specie(id,nome, idRisorsa)

RisorsaMedica(id, nome, prezzo,idFornitore)

RisorsaAlimentare(id, nome, prezzo,dataScadenza , idFornitore)

Fornitore(id, nome)

QUERY E PROCEDURE

Query che ritorna l'id e il nome dei fornitori che forniscono risorse alimentari e mediche

```
select distinct f.id, f.nome
from Fornitore f
where exists (
    select *
    from Fornitore f1 join RisorsaAlimentare a on (f1.id=a.idFornitore)
    where f.id=f1.id and a.idFornitore in (
        select idFornitore
        from RisorsaMedica
    )
);
```

OUTPUT

```
+-----+-----+
| id | nome      |
+-----+-----+
| 902 | cocacola.Srl |
+-----+-----+
```

Query che ritorna il nome dei fornitori che forniscono solo risorse alimentari dei quali prodotti non scadono prima di marzo.

```
select distinct f.nome
from Fornitore f join RisorsaAlimentare m on (f.id = m.idFornitore)
where m.dataScadenza >= '2019-03-01' and f.id not in (
    select idFornitore
    from RisorsaMedica
);
```

OUTPUT

```
+-----+
| nome   |
+-----+
| pepsi.Co |
| stanco.Co |
+-----+
```

Query che ritorna i nomi e cognomi dei clienti che hanno effettuato una prenotazione, nelle quali sono state assegnate diversi impiegati.

```
select c.nome, c.cognome
from Cliente c
where exists(
    select *
    from Prenotazione p join Impiegato i on ( p.idImpiegato = i.id)
    where c.email = p.email and p.idImpiegato <> any (
        select p1.idImpiegato
        from Prenotazione p1 join Impiegato i1 on ( p1.idImpiegato = i1.id)
        where p.email = p1.email
    )
);
```


OUTPUT

nome	cognome
Blaise	Matuidi
Cristiano	Ronaldo
Douglas	Costa
Federico	Bernardeschi
Leonardo	Bonucci

Query che ritorna il nome e quante volte una risorsa è stata usata in nelle visite.

```
select r.nome, count(*) as numeroUsi
from RisorsaMedica r, Visita v
where r.id = v.medica
group by r.nome;
```

OUTPUT

nome	numeroUsi
medicine	6
siringa	7

Procedura che restituisce il numero di volpi su cui un dato impiegato ha effettuato una visita durante l'anno scorso.

```
delimiter $
create procedure NumeroVisite( Id_Impiegato int)
begin
select count(*) as Visite
from Visita
where idImpiegato = Id_Impiegato and year(dataVisita) = year(now()) - 1;
end $
```

```
call NumeroVisite(11);
```

OUTPUT

```
+-----+
| Visite |
+-----+
|    3   |
+-----+
```

Procedura che ritorna il numero di volpi che sono sorvegliate da un dato impiegato.

```
create procedure volpi_settore(Id_Impiegato int)
begin
select count(*) as Numero
from  Volpe v1 join Specie s1 on (v1.idSpecie = s1.id ) join Settore r on ( s1.id
= r.specie ) join Guardia on (r.nome = settore)
where idImpiegato = Id_Impiegato;
end $
```

```
call volpi_settore(1);
```

OUTPUT

```
+-----+
| Numero |
+-----+
|    4   |
+-----+
```

Funzione che ritorna il prezzo complessivo di una data risorsa medica

```
CREATE FUNCTION PrezzoTotaleRisorsa (Id_ris INT)
RETURNS INT
BEGIN
DECLARE Totale INT ;
SELECT sum(r.prezzo) INTO Totale
FROM RisorsaMedica r JOIN Visita v ON(r.id= v.medica)
where Id_ris = id
GROUP BY r.id;
RETURN Totale;
END $
```

Funzione che restituisce il numero di volpe che mangiano lo stesso alimento

```
create function mangiaVolpi (Id_Alimento int)
returns int
begin
declare num int;
select count(*) into num
from Volpe v join Specie s on ( v.idSpecie = s.id)
where s.idRisorsa = Id_Alimento;
return num;
end $
```

Trigger che controlla la qualità di un alimento quando viene aggiornato

```
create trigger ControlloQualita
after update on RisorsaAlimentare
for each row
begin
if ( new.dataScadenza > curdate())
then
update RisorsaAlimentare
set dataScadenza = new.dataScadenza
where dataScadenza = old.dataScadenza;
end if;
end $
```

Trigger che prima dell'inserimento nel database di una risorsa ,inserisce nella tabella fornitore l' id del fornitore se non esiste già.

```
CREATE TRIGGER RegistraMedica
BEFORE INSERT ON RisorsaMedica
FOR EACH ROW
BEGIN
IF new.idFornitore NOT IN (
SELECT F.id
From Fornitore F )
THEN
insert into Fornitore (id) values(new.idFornitore);
END IF;
END $
```

```
CREATE TRIGGER RegistraAlimento
BEFORE INSERT ON RisorsaAlimentare
FOR EACH ROW
BEGIN
IF new.idFornitore NOT IN (
SELECT F.id
From Fornitore F )
THEN
insert into Fornitore (id values(new.idFornitore);
END IF;
END $
```