Team Name- data worms

Members:
Tirth Motka
Jay Ghevariya
Vidhi Pareek

Crufts ER Model

Introduction

The mini world is about Crufts, an international dog show (competition-based).
This is a database of a dog show, which deals with all the functioning and maintenance of records related to all aspects of the show. It contains all the information about the show (i.e from location of the show to records of judges and winners).

Purpose

Without the database , it will be very difficult to keep record of all past and upcoming events. The database makes it easy to maintain all the records. Some examples include keeping record of location and other details of all the past shows, storing the details of all the dogs and their owners etc.

Users

The users of the database will be:

1. Event organising team
2. Owners of the pets who registered for a show
3. Ticket buyers

Applications

This database will keep a record of :

1. Owners and their pets who registered for any past competition or for an event that will happen in future.
2. Past dog shows and their winners
3. Details of all past and to be organised events
4. Record of ticket buyers for the shows (Buy Ticket option available on the website).

All the required functionalities (like inserting details of new shows , registering new dogs etc) will be facilitated.

Database Requirements

Entities

1)Owner

A. Owner_Id (Primary Key) (BIGINT(9,223,372,036, 854,775,807) and Owner_Id>0)
B. Event_Id (Foreign Key) (BIGINT(9,223,372,036, 854,775,807) and Event_Id>0)
C. First Name (VARCHAR(100))
D. Last Name (VARCHAR(100))
E. Address (TEXT(1000))
F. E-mail Id (VARCHAR(100))
G. Contact Number (VARCHAR(15))
H. Date of Registration (DATE)

2) Events

A. Event_Id (Primary Key) (BIGINT(9,223,372,036, 854,775,807) and Event_Id>0)
B. Name (VARCHAR(100))
C. Location(TEXT(1000))
D. Date (DATE)
E. Timings (TIME)
F. Seats Available (INT(1000000)) *This can be decided as per organisational requirement

3) Judges

A. Event_Id(Foreign Key) (BIGINT(9,223,372,036, 854,775,807) and Event_Id>0)
B. First Name (VARCHAR(100))
C. Last Name (VARCHAR(100))
D. Address (TEXT(1000))
E. E-mail Id (VARCHAR(100))
F. Contact Number (Primary Key)(VARCHAR(15))
G. Category (VARCHAR(100))

4) Ticket Buyers

A. Event_Id (Foreign Key)(BIGINT(9,223,372,036, 854,775,807) and Event_Id>0)
B. First Name (VARCHAR(100))
C. Last Name (VARCHAR(100))
D. Address (TEXT(1000))
E. E-mail Id (Primary Key) (VARCHAR(100))

F. Contact Number (VARCHAR(15))
G. Date of Registration (DATE)
H. Status of Payment (TEXT(8) : “Paid” or “Pending”)

Weak Entity:

1) Dog (Primary Key = Owner_Id + Name)

A. Owner_Id (Foreign Key) (BIGINT(9,223,372,036, 854,775,807) and Owner_Id>0)
B. Category (VARCHAR(100))
C. Name (VARCHAR(100))
D. Breed (VARCHAR(100))
E. Gender (CHAR(1) : “M” or “F”)
F. Date of Birth (DATE)
G. Age (INT(20))

2) Winners (Primary Key = Owner_Id + Name)

A. Owner_Id (Foreign Key) (BIGINT(9,223,372,036, 854,775,807) and Owner_Id>0)
B. Name (VARCHAR(100))
C. Category (VARCHAR(100))

Relationships

The relationships that exist between the entities are as follows:

1. Dog ---> Owner Common Attribute: Owner_Id {cardinality: 1:1}
2. Winners -> Dog : common attribute ——> Owner_Id {cardinality: 1:1}
3. Winners->Owner: common attribute ——> Owner_Id {cardinality: 1:1}
4. Judges->Events:common attribute ——>Event_Id {cardinality: 1:1}
5. Ticket buyers-> Events:common attribute ——>Event_Id { cardinality: N:M}
6. Owner-> Events:common attribute ——>Event_Id { cardinality: N:1}

n>=3 Relationships:

List all the n>=3 relationships.

1.Ticket Buyers -> Events -> Judge -> Owner -> Dog -> Winner

- This is a n>=3 relationship which comprises many smaller relationships like Event->Judge->Owner->Winner , Event->Judge->Owner->Dog , Ticket Buyers ->Events -> Owner->Winner etc.

Functional Requirements

Modifications

1. Insert: Inserting the details of owner , dog , events , judges , ticket buyers.
   1. for Owner -> Owner_Id,Event_Id, First Name, Last Name, Address, E-mail Id, Contact Number,Date of Registration
   2. for Events-> Events_Id,Name, Location,Date,Timings,Seats Available
   3. for Judge-> Event_Id, First Name,Last Name,Address,E-mail Id,Contact Number,Category
   4. for Ticket Buyers-> Event_Id,First Name,Last Name,Address,E-mail Id,Contact Number,Date of Registration,Status of Payment
   5. for Dog-> Owner_Id ,Category,name,Breed,gender,Date of Birth , Age
   6. for Winners-> Owner_Id,Name,Category

2. Delete: If an owner withdraws his/her pet , then the owner and dog data entries have to be removed or if someone cancels tickets for the show.

3. Update: For an upcoming event, as the tickets are booked Seats Available have to be updated in Event.

Retrievals

1. Selection:
   Examples :
   1.Select all the dogs of a particular breed (like English Toy Terrier)
   2. Select all the owners with last name "Oliver"
2. Projection:
   Example :
   1. The owners of dog whose age >= 1 year
   2. The winners whose age <=2 year
2. Aggregate:
   Examples :
   1. The highest age of any dog
   2. Average age of all winners
5. Search:
   Examples :
   1. Output the names of owners beginning with “Joh” (John , Jonathan etc.)
3. Analysis:

1. Generate a report which contains details of owners whose pets got the award for Best In Show.
2. Generate a report which contains details of the winners of all past events.

Some assumptions during this model:
1. There will only be one owner to one dog and one dog per owner.
2. One owner-dog pair can participate in only one event .
3. There will be unique judge for every event .