



# Registro de Actividades Hípicas

---

**Materia:** 75.15 Bases de Datos

**Grupo:** 10

**Fecha de entrega:** 29 de mayo del 2013

**Integrantes:**

- Fraiman, Yamila (90708)
- Goldberg, Juan Sebastián (82078)
- Mezzina, Agustín (89637)
- Ramos, Milton (91110)



## Contents

Hipotesis y Supuestos .....	2
Validaciones a realizar.....	2
Diagrama Entidad Interrelación .....	3
Modelo Relacional.....	4
Diccionario de datos.....	6

## Hipotesis y Supuestos

1. Puede existir más de un dueño por equino.
2. Puede existir más de un cuidador por equino.
3. Puede existir más de un entrenador por equino.
4. La madre y/o el padre de un equino pueden ser desconocidos.

## Validaciones a realizar

1. Un equino solo puede participar en una única carrera por encuentro.
2. Una carrera puede ser solo de caballos o solo de yeguas.
3. En una carrera solo pueden correr equinos que clasifiquen según el tipo de carrera definido.
4. Para todo Período, habrá que validar que no exista solapamiento entre fechas de una misma relación.
5. Un jockey debe utilizar por carrera un gorro y una chaquetilla que pertenezca al stud del caballo que monta.
6. Un equino no puede ser antecesor de sí mismo en el árbol genealógico.
7. La madre de un equino debe ser hembra y el padre macho.



## Diagrama Entidad Interrelación



## Modelo Relacional

A continuación se definen las relaciones correspondientes al modelo presentado en los diagramas entidad interrelación.

Por cada relación se indican sus campos y su PK. Luego, por relación se indican las FKs y validaciones a realizar:

- Hipodromo(NombreHipodromo)
- Pista(NumeroPista, NombreHipodromo, TipoPista, CantidadAndariveles)
  - FKs:
    - (NombreHipodromo)
- Andarivel(NumeroPista, NumeroAndarivel)
  - FKs:
    - (NumeroPista)
  - Validaciones:
    - NumeroAndarivel debe ser mayor igual a 1 y menor igual a CantidadAndariveles de la pista a la que pertenece.
- Encuentro(FechaEncuentro, NombreHipodromo)
  - FKs:
    - (NombreHipodromo)
- CondicionInscripción(IdCondicionInscripcion, DescripcionCondicion)
- Carrera(FechaEncuentro, NumeroCarrera, NumeroPista, GeneroCarrera, TipoCarrera, DistanciaCarrera, EstadoTiempo, EstadoPista, FechaLimiteInscripcion, IdCondicionInscripcion, HoraCarrera)
  - FKs:
    - (FechaEncuentro)
    - (NumeroPista)
    - (IdCondicionInscripcion)
  - Validaciones:
    - El hipódromo del encuentro debe coincidir con el hipódromo de la pista
- Pelaje(NombrePelaje)
- Equino(NumeroEquino, Genero, NombrePelaje, FechaNacimiento, Peso)
  - FKs:
    - (NombrePelaje)
- HijoDe(NumeroEquino, Padre, Madre)
  - FKs:
    - (Hijo): (NumeroEquino)
    - (Padre): (NumeroEquino)
    - (Madre): (NumeroEquino)
  - Validaciones:
    - Un equino no puede ser antecesor de sí mismo en el árbol genealógico.
    - La madre debe ser una hembra.
    - El padre debe ser un macho.
- Stud(NombreStud)
- Color(Color)
- ColorStud(NombreStud, Color)
  - FKs:
    - (NombreStud)
    - (Color)
- Gorro(DiseñoGorro, NombreStud)
  - FKs:
    - (NombreStud)
- ColorGorro(DiseñoGorro, Color)
  - FKs:
    - (NombreStud, DiseñoGorro)
    - (Color)
- Chaquetilla(DiseñoChaquetilla, NombreStud)
  - FKs:
    - (NombreStud)
- ColorChaquetilla(DiseñoChaquetilla, Color)
  - FKs:
    - (NombreStud, DiseñoChaquetilla)
    - (Color)
- StudEquino(NumeroEquino, FechaDesde, FechaHasta, NombreStud)



- FKs:
  - (NumeroEquino)
- Validaciones:
  - No puede haber solapamiento de periodos (FechaDesde, FechaHasta) para un mismo NumeroEquino.
- ParticipacionCarrera(FechaEncuentro, NumeroCarrera, NumeroAndarivel, TiempoCarrera, LastreJockey, NumeroEquino, NumeroLicenciaJockey, TipoLicenciaJockey, DiseñoChaquetilla, DiseñoGorro, FechaInscripcion, DiferenciaCuerpos, PesoJockey, PesoEquino)
  - FKs:
    - (FechaEncuentro, NumeroCarrera)
    - (NumeroEquino)
    - (NumeroLicenciaJockey, TipoLicenciaJockey)
    - (DiseñoChaquetilla)
    - (DiseñoGorro)
  - Validaciones:
    - En un encuentro (el encuentro al que pertenece la carrera) un equino solo puede participar una única vez.
    - En una carrera (FechaEncuentro, NumeroCarrera) solo pueden participar equinos (NumeroEquino) del mismo género.
    - En una carrera (FechaEncuentro, NumeroCarrera) solo pueden participar equinos (NumeroEquino) para los que se corresponda su Handicap.
    - El Stud del equino, de la chaquetilla y del gorro deben coincidir.
    - El número de andarivel debe existir para la pista en la que se realiza la carrera.
    - Se debe validar que se cumplan los requisitos descriptos según la condición de inscripción (IdCondicionInscripcion) de la carrera (FechaEncuentro, NumeroCarrera).
- Persona(DNI, Nombre, Apellido)
- Rol(NumeroRol, DNI, FechaDesde, FechaHasta)
  - FKs:
    - (DNI)
- Propietario(NumeroRol)
  - FKs:
    - (NumeroRol)
  - Validaciones:
    - Para un mismo DNI, no debe haber solapamiento de periodos en la relación Rol.
- Cuidador(NumeroRol)
  - FKs:
    - (NumeroRol)
  - Validaciones:
    - Para un mismo DNI, no debe haber solapamiento de periodos en la relación Rol.
- Entrenador(NumeroRol, NumeroLicenciaEntrenador)
  - FKs:
    - (NumeroRol)
  - Validaciones:
    - Para un mismo DNI, no debe haber solapamiento de periodos en la relación Rol.
    - Para distintos DNI no puede haber un mismo NumeroLicenciaEntrenador.
- Categoria(NombreCategoria)
- Jockey(NumeroRol, TipoLicenciaJockey, NumeroLicenciaJockey, Peso, NombreCategoria, HipodromoLicencia)
  - FKs:
    - (NumeroRol)
    - (NombreCategoria)
    - (HipodromoLicencia)
  - Validaciones:
    - Para un mismo DNI, no debe haber solapamiento de periodos en la relación Rol.
    - Para distintos DNI no puede haber un mismo (TipoLicenciaJockey, NumeroLicenciaJockey).
- Posee(NumeroRol, NumeroEquino, FechaDesde, FechaHasta)
  - FKs:
    - (NumeroRol) (respecto de la relación Propietario)
    - (NumeroEquino)
  - Validaciones:
    - El período de posesión debe encontrarse dentro del período del rol.
    - Para un mismo (DNI, NumeroEquino) no debe haber solapamiento de periodos.
- Cuida(NumeroRol, NumeroEquino, FechaDesde, FechaHasta)
  - FKs:
    - (NumeroRol) (respecto de la relación Cuidador)
    - (NumeroEquino)
  - Validaciones



- El período de cuidado debe encontrarse dentro del período del rol.
  - Para un mismo (DNI, NumeroEquino) no debe haber solapamiento de períodos.
- Entrena(NumeroRol, NumeroEquino, FechaDesde, FechaHasta)
  - FKs:
    - (NumeroRol) (respecto de la relación Entrenador)
    - (NumeroEquino)
  - Validaciones
    - El período de entrenamiento debe encontrarse dentro del período del rol.
    - Para un mismo (DNI, NumeroEquino) no debe haber solapamiento de períodos.

## Diccionario de datos

```
CREATE TABLE Hipodromo (
  NombreHipodromo varchar(100) NOT NULL,
  PRIMARY KEY (NombreHipodromo)
);

CREATE TABLE Pista (
  NombreHipodromo varchar(100) NOT NULL,
  NumeroPista tinyint NOT NULL,
  TipoPista tinyint NOT NULL,
  CantidadAndariveles tinyint NOT NULL,
  PRIMARY KEY (NombreHipodromo, NumeroPista),
  CONSTRAINT FK_Pista_NombreHipodromo FOREIGN KEY (NombreHipodromo) REFERENCES
Hipodromo(NombreHipodromo)
);

CREATE TABLE Andarivel (
  NombreHipodromo varchar(100) NOT NULL,
  NumeroPista tinyint NOT NULL,
  NumeroAndarivel tinyint NOT NULL,
  PRIMARY KEY (NombreHipodromo, NumeroPista, NumeroAndarivel),
  CONSTRAINT FK_Andarivel_NombreHipodromoNumeroPista FOREIGN KEY (NombreHipodromo,
NumeroPista) REFERENCES Pista(NombreHipodromo, NumeroPista)
);

CREATE TABLE Encuentro (
  NombreHipodromo varchar(100) NOT NULL,
  FechaEncuentro date NOT NULL,
  NumeroEncuentro int NOT NULL,
  PRIMARY KEY (NombreHipodromo, FechaEncuentro),
  CONSTRAINT FK_Encuentro_NombreHipodromo FOREIGN KEY (NombreHipodromo) REFERENCES
Hipodromo(NombreHipodromo)
);

CREATE TABLE Carrera (
  NombreHipodromo varchar(100) NOT NULL,
  FechaEncuentro date NOT NULL,
  NumeroPista tinyint NOT NULL,
  NumeroCarrera int NOT NULL,
  GeneroCarrera varchar(1) NULL,
  TipoCarrera varchar(100) NULL,
  DistanciaCarrera int NULL,
  EstadoTiempo varchar(255) NULL,
  EstadoPista varchar(255) NULL,
  PRIMARY KEY (NombreHipodromo, FechaEncuentro, NumeroPista, NumeroCarrera),
  CONSTRAINT FK_Carrera_NombreHipodromoFechaEncuentro FOREIGN KEY (NombreHipodromo,
FechaEncuentro) REFERENCES Encuentro(NombreHipodromo, FechaEncuentro),
  CONSTRAINT FK_Carrera_NombreHipodromoNumeroPista FOREIGN KEY (NombreHipodromo,
NumeroPista) REFERENCES Pista(NombreHipodromo, NumeroPista)
);

CREATE TABLE Pelaje (
  NombrePelaje varchar(100) NOT NULL,
  PRIMARY KEY (NombrePelaje)
);
```



```
CREATE TABLE Equino (
    NumeroEquino int NOT NULL,
    Genero varchar(1) NULL,
    NombrePelaje varchar(100) NULL,
    PRIMARY KEY (NumeroEquino),
    CONSTRAINT FK_Equino_NombrePelaje FOREIGN KEY (NombrePelaje) REFERENCES
Pelaje(NombrePelaje)
);

CREATE TABLE HijoDe (
    NumeroEquino int NOT NULL,
    Padre int NULL,
    Madre int NULL,
    PRIMARY KEY (NumeroEquino),
    CONSTRAINT FK_HijoDe_NumeroEquino FOREIGN KEY (NumeroEquino) REFERENCES
Equino(NumeroEquino),
    CONSTRAINT FK_HijoDe_Padre FOREIGN KEY (Padre) REFERENCES Equino(NumeroEquino),
    CONSTRAINT FK_HijoDe_Madre FOREIGN KEY (Madre) REFERENCES Equino(NumeroEquino)
);

CREATE TABLE Stud (
    NombreStud varchar(100) NOT NULL,
    PRIMARY KEY (NombreStud)
);

CREATE TABLE Color (
    Color varchar(6) NOT NULL,
    PRIMARY KEY (Color)
);

CREATE TABLE ColorStud (
    NombreStud varchar(100) NOT NULL,
    Color varchar(6) NOT NULL,
    PRIMARY KEY (NombreStud, Color),
    CONSTRAINT FK_ColorStud_NombreStud FOREIGN KEY (NombreStud) REFERENCES Stud(NombreStud),
    CONSTRAINT FK_ColorStud_Color FOREIGN KEY (Color) REFERENCES Color(Color)
);

CREATE TABLE Gorro (
    NombreStud varchar(100) NOT NULL,
    DisenoGorro varchar(100) NOT NULL,
    PRIMARY KEY (NombreStud, DisenoGorro),
    CONSTRAINT FK_Gorro_NombreStud FOREIGN KEY (NombreStud) REFERENCES Stud(NombreStud)
);

CREATE TABLE ColorGorro (
    NombreStud varchar(100) NOT NULL,
    DisenoGorro varchar(100) NOT NULL,
    Color varchar(6) NOT NULL,
    PRIMARY KEY (NombreStud, DisenoGorro, Color),
    CONSTRAINT FK_ColorGorro_NombreStudDisenoGorro FOREIGN KEY (NombreStud, DisenoGorro)
REFERENCES Gorro(NombreStud, DisenoGorro),
    CONSTRAINT FK_ColorGorro_Color FOREIGN KEY (Color) REFERENCES Color(Color)
);

CREATE TABLE Chaquetilla (
    NombreStud varchar(100) NOT NULL,
    DisenoChaquetilla varchar(100) NOT NULL,
    PRIMARY KEY (NombreStud, DisenoChaquetilla),
    CONSTRAINT FK_Chaquetilla_NombreStud FOREIGN KEY (NombreStud) REFERENCES Stud(NombreStud)
);

CREATE TABLE ColorChaquetilla (
    NombreStud varchar(100) NOT NULL,
    DisenoChaquetilla varchar(100) NOT NULL,
    Color varchar(6) NOT NULL,
    PRIMARY KEY (NombreStud, DisenoChaquetilla, Color),
    CONSTRAINT FK_ColorChaquetilla_NombreStudDisenoChaquetilla FOREIGN KEY (NombreStud,
DisenoChaquetilla) REFERENCES Chaquetilla(NombreStud, DisenoChaquetilla),
    CONSTRAINT FK_ColorChaquetilla_Color FOREIGN KEY (Color) REFERENCES Color(Color)
);
```



```
);

CREATE TABLE StudEquino (
    NumeroEquino int NOT NULL,
    FechaDesde date NOT NULL,
    FechaHasta date NULL,
    NombreStud varchar(100) NULL,
    PRIMARY KEY (NumeroEquino, FechaDesde),
    CONSTRAINT FK_StudEquino_NumeroEquino FOREIGN KEY (NumeroEquino) REFERENCES
Equino(NumeroEquino)
);

CREATE TABLE ParticipacionCarrera (
    NombreHipodromo varchar(100) NOT NULL,
    FechaEncuentro date NOT NULL,
    NumeroPista tinyint NOT NULL,
    NumeroCarrera int NOT NULL,
    NumeroAndarivel tinyint NOT NULL,
    TiempoCarrera double NULL,
    NumeroEquino int NULL,
    NumeroLicenciaJockey int NOT NULL,
    TipoLicenciaJockey varchar(100) NOT NULL,
    DiseÑoChaquetilla varchar(100) NULL,
    DiseÑoGorro varchar(100) NULL,
    LastreJockey tinyint NOT NULL DEFAULT 0,
    INDEX (NumeroLicenciaJockey, TipoLicenciaJockey),
    PRIMARY KEY (NombreHipodromo, FechaEncuentro, NumeroPista, NumeroCarrera,
NumeroAndarivel),
    CONSTRAINT FK_ParticipacionCarrera_Carrera FOREIGN KEY (NombreHipodromo, FechaEncuentro,
NumeroPista, NumeroCarrera) REFERENCES Carrera(NombreHipodromo, FechaEncuentro, NumeroPista,
NumeroCarrera),
    CONSTRAINT FK_ParticipacionCarrera_Andarivel FOREIGN KEY (NombreHipodromo, NumeroPista,
NumeroAndarivel) REFERENCES Andarivel(NombreHipodromo, NumeroPista, NumeroAndarivel),
    CONSTRAINT FK_ParticipacionCarrera_NumeroEquino FOREIGN KEY (NumeroEquino) REFERENCES
Equino(NumeroEquino),
    CONSTRAINT FK_ParticipacionCarrera_Jockey FOREIGN KEY (NumeroLicenciaJockey,
TipoLicenciaJockey) REFERENCES Jockey(NumeroLicenciaJockey, TipoLicenciaJockey)
);

CREATE TABLE Persona (
    DNI int NOT NULL,
    Nombre varchar(100) NOT NULL,
    Apellido varchar(100) NOT NULL,
    PRIMARY KEY (DNI)
);

CREATE TABLE Rol (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    FechaDesde date NOT NULL,
    FechaHasta date NULL,
    PRIMARY KEY (DNI, NumeroRol),
    CONSTRAINT FK_Rol_DNI FOREIGN KEY (DNI) REFERENCES Persona(DNI)
);

CREATE TABLE Propietario (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    PRIMARY KEY (DNI, NumeroRol),
    CONSTRAINT FK_Propietario_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Rol(DNI,
NumeroRol)
);

CREATE TABLE Cuidador (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    PRIMARY KEY (DNI, NumeroRol),
    CONSTRAINT FK_Cuidador_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Rol(DNI,
NumeroRol)
);
```





```
CREATE TABLE Entrenador (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    NumeroLicenciaEntrenador int NOT NULL,
    PRIMARY KEY (DNI, NumeroRol),
    CONSTRAINT FK_Entrenador_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Rol(DNI,
NumeroRol)
);

CREATE TABLE Categoria (
    NombreCategoria varchar(100) NOT NULL,
    PRIMARY KEY (NombreCategoria)
);

CREATE TABLE Jockey (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    TipoLicenciaJockey varchar(100) NOT NULL,
    NumeroLicenciaJockey int NOT NULL,
    Peso double NOT NULL,
    NombreCategoria varchar(100) NOT NULL,
    PRIMARY KEY (DNI, NumeroRol),
    INDEX (NumeroLicenciaJockey, TipoLicenciaJockey),
    CONSTRAINT FK_Jockey_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Rol(DNI,
NumeroRol),
    CONSTRAINT FK_Jockey_NombreCategoria FOREIGN KEY (NombreCategoria) REFERENCES
Categoria(NombreCategoria)
);

CREATE TABLE Possee (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    NumeroEquino int NOT NULL,
    PRIMARY KEY (DNI, NumeroRol, NumeroEquino),
    CONSTRAINT FK_Posee_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Propietario(DNI,
NumeroRol),
    CONSTRAINT FK_Posee_NumeroEquino FOREIGN KEY (NumeroEquino) REFERENCES Equino
(NumeroEquino)
);

CREATE TABLE Cuida (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    NumeroEquino int NOT NULL,
    PRIMARY KEY (DNI, NumeroRol, NumeroEquino),
    CONSTRAINT FK_Cuida_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Cuidador(DNI,
NumeroRol),
    CONSTRAINT FK_Cuida_NumeroEquino FOREIGN KEY (NumeroEquino) REFERENCES Equino
(NumeroEquino)
);

CREATE TABLE Entrena (
    DNI int NOT NULL,
    NumeroRol int NOT NULL,
    NumeroEquino int NOT NULL,
    PRIMARY KEY (DNI, NumeroRol, NumeroEquino),
    CONSTRAINT FK_Entrena_DNINumeroRol FOREIGN KEY (DNI, NumeroRol) REFERENCES Entrenador(DNI,
NumeroRol),
    CONSTRAINT FK_Entrena_NumeroEquino FOREIGN KEY (NumeroEquino) REFERENCES Equino
(NumeroEquino)
);
```