

Fundamentos de Bases de Datos. Septiembre 2015

Ejercicio Práctico

Nombre Alumno:

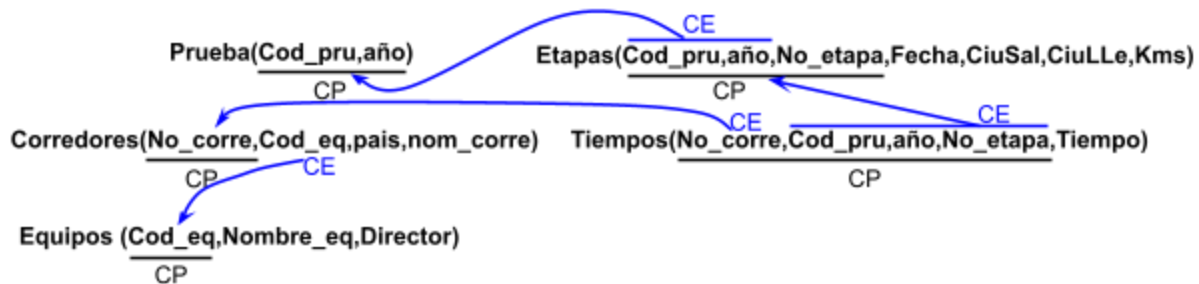
Profesor del Grupo:

Parcial/es a los que se presenta:

Conserva nota participacion?:

Parcial 2

Disponemos de la siguiente BD que gestiona la la información relativa a competiciones ciclisticas por etapas.



Se pide:

- A. Escribe las instrucciones en SQL para la creación de la tabla Tiempos y la inserción de una tupla en dicha tabla, suponiendo que el campo `Tiempo` es de tipo numérico entero positivo y que almacena los segundos invertidos por un corredor en una etapa. Para la creación de la tabla se presuponen creadas el resto de tablas. Además de las restricciones de integridad especificadas en el esquema, deben considerarse las siguientes:
- Los campos `No_corre` y `No_etapa` son de tipo entero positivo mayores que 0 y con un valor máximo de 9999 para el primero y de 30 para el segundo. El campo `Tiempo` siempre debe contener un valor. El campo `año` toma un valor entero positivo de cuatro dígitos comprendido mayor o igual que 1900 y el campo `Cod_pru` toma valores alfanuméricos con un máximo de 30 caracteres. (1.25 pts.).

Res:

```
CREATE TABLE Tiempos(  
  No_corre Number(4) CHECK(No_corre BETWEEN 1 AND 9999)  
    REFERENCES Corredores(No_corre) ,  
  Cod_pru varchar(30) ,  
  año number(4) CHECK (año >=1900) ,  
  No_etapa number(2) CHECK(No_etapa BETWEEN 1 AND 30) ,  
  Tiempo number NOT NULL CHECK(Tiempo >0) ,  
  FOREIGN KEY (Cod_pru,año,No_etapa)  
    REFERENCES Etapas(Cod_pru,año,No_etapa) ,  
  PRIMARY KEY(No_corre,Cod_pru,año,No_etapa)  
);
```

```
INSERT INTO Tiempos VALUES(1,'VueltaEpaña',2014,12,20000);
```

B. Realiza las siguientes consultas::

a. “Muestra la etapa más corta”.(en **AR 1.25 pts.** y **CRT 1.25 pts.**).

$\rho(Etapas) = E1$

$\rho(Etapas) = E2$

$Etapas - \pi_{E1.Cod_pru,E1.año,E1.No_etapa,E1.Fecha,E1.CiuSal,E1.CiuLle,E1.kms}(\sigma_{E1.Kms>E2.Kms}(E1 \times E2))$

```
{E1 | etapas(E1) and not (exists E2) (etapas(E2) and E1.Kms>E2.Kms) };
```

```
{E1 | etapas(E1) and (forall E2) (etapas(E2) and E1.Kms<=E2.Kms) };
```

b. “Muestra el nombre de los corredores que han realizado todas las etapas de la prueba ‘Giro de Italia’ del 2015”. (en **AR 1.25pts.** y **SQL 1.25 pts.**)

$\pi_{Nom_corre}(Corredores \bowtie (\pi_{No_corre,Cod_pru,año,No_etapa}(Tiempos) \div \pi_{Cod_pru,año,No_etapa}(\sigma_{cod_pru='Giro de Italia' \wedge año=2015}(Etapas))))$

```
Select C.nom_corre From corredores C Where not exists(
Select * From Etapas E Where E.cod_pru='Giro de Italia' and E.año=2015
and not exists(
Select * From Tiempos T Where T.no_corre=C.no_corre and
T.Cod_pru=E.cod_pru and T.año=E.año and T.No_etapa=E.No_etapa
)
);
```

```
Select C.nom_corre From corredores C Where not exists(
Select Cod_pru,año,No_etapa From Etapas E Where E.cod_pru='Giro de
Italia' and E.año=2015
Minus
Select Cod_pru,año,No_etapa From Tiempos T Where T.no_corre=C.no_corre
)
);
```

c. “Muestra los equipos cuyos corredores son todos del mismo país”. (en **CRT 1.25 pts.** y **SQL 1.25 pts.**).

```
{C1.cod_eq | corredores(C1) and not (exists C2) (corredores(C2) and
C1.cod_eq=C2.cod_eq and C1.pais<>C2.pais) };
```

```
Select C1.cod_eq From corredores C1 Where not exists(
Select * From corredores C2 Where C1.cod_eq=C2.cod_eq and
C1.pais<>C2.pais
)
```

- C. Crea una vista que muestre la clasificación general individual del 'Tour de Francia' del 2014 (nombre del corredor y tiempo total invertido, ordenada de forma creciente con respecto al tiempo total empleado). **(1.25 pt.)**.

```
Create View GeneralTour2014 AS
Select T.No_corre,sum(T.tiempo) From Tiempos T
WHERE T.cod_pru='Tour de Francia' and año=2014
Group by T.No_corre
Order by sum(T.tiempo) Desc;
```

Tiempo de realización: **Un Parcial 1.5h, dos parciales 3h.**