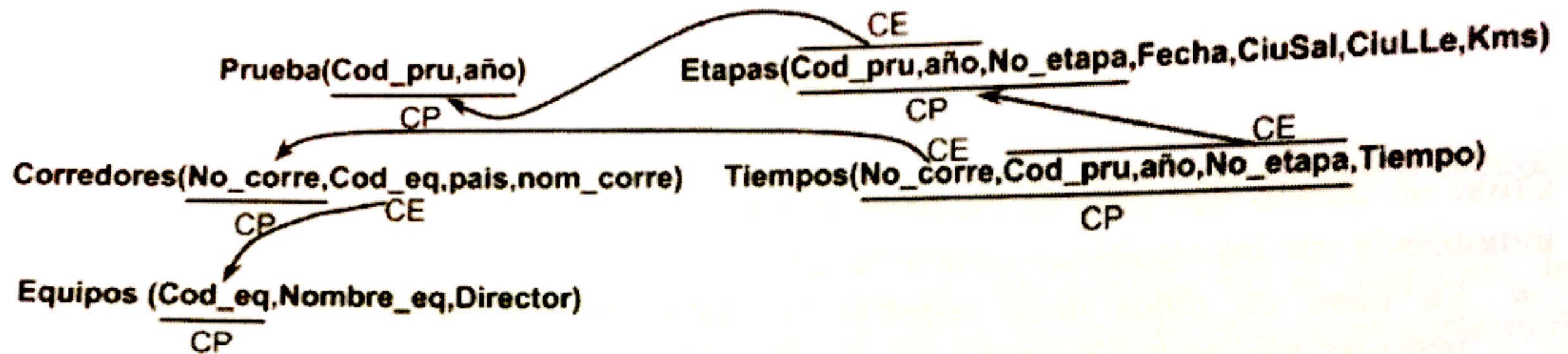
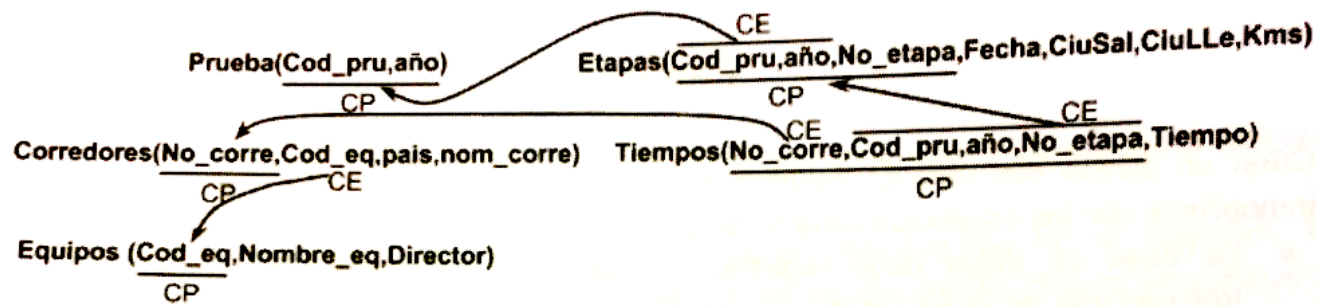


Parcial 2

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





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 mayor o igual que 1900 y el campo Cod_pru toma valores alfanuméricos con un máximo de 30 caracteres. (1.25 pts.).

```

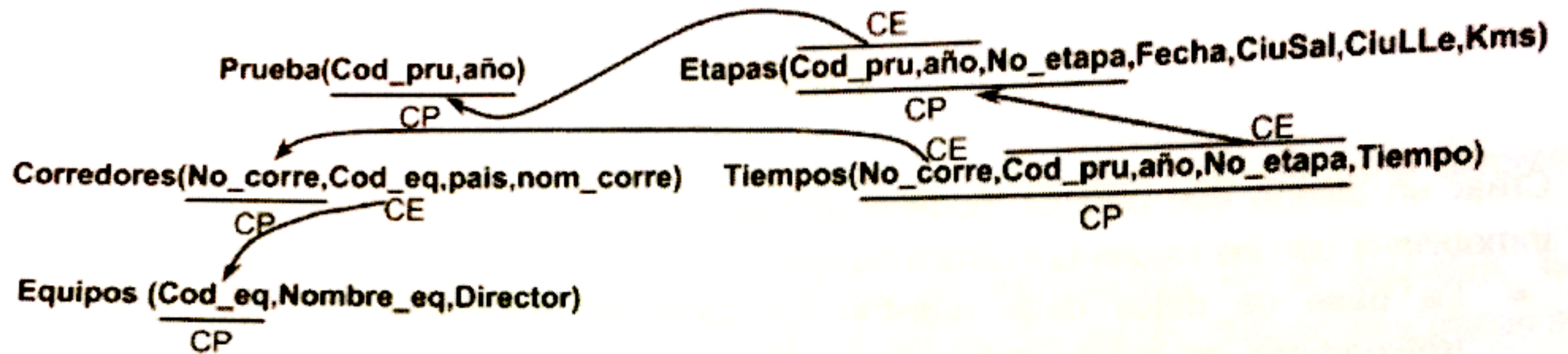
CREATE TABLE Tiempos (
  No_corre INT CHECK((No_corre > 0) AND (No_corre <= 9999)) REFERENCES Corredores(No_corre),
  Cod_pru VARCHAR2(30),
  año NUMBER(4) CHECK(año >= 1900),
  No_etapa INT CHECK((No_etapa > 0) AND (No_etapa <= 30)),
  Tiempo INT NOT NULL CHECK(Tiempo >= 0),
  FOREIGN KEY (Cod_pru,año,No_etapa) REFERENCES Etapas(Cod_pru,año,No_etapa),
  PRIMARY KEY (No_corre,Cod_pru,año,No_etapa)
);

```

```

INSERT INTO Tiempos VALUES (1, 'Prueba', 2021, 1, 60);

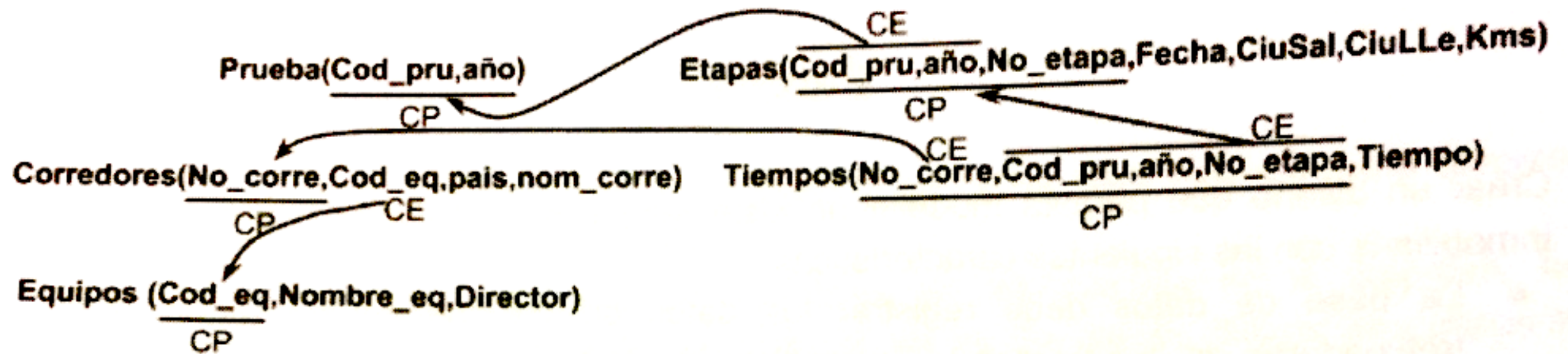
```



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

$$\pi_{\text{Cod_pru, anno, No_etapa}}(\text{Etapas}) - \pi_{\text{A.Cod_pru, A.anno, A.No_etapa}}(\sigma_{\text{A.Kms} > \text{B.Kms}}(\rho_{\text{A}}(\text{Etapas}) \times \rho_{\text{B}}(\text{Etapas})))$$

```
SELECT Cod_pru, año, No_etapa FROM Etapas WHERE Kms =
(SELECT MIN(Kms) FROM Etapas);
```

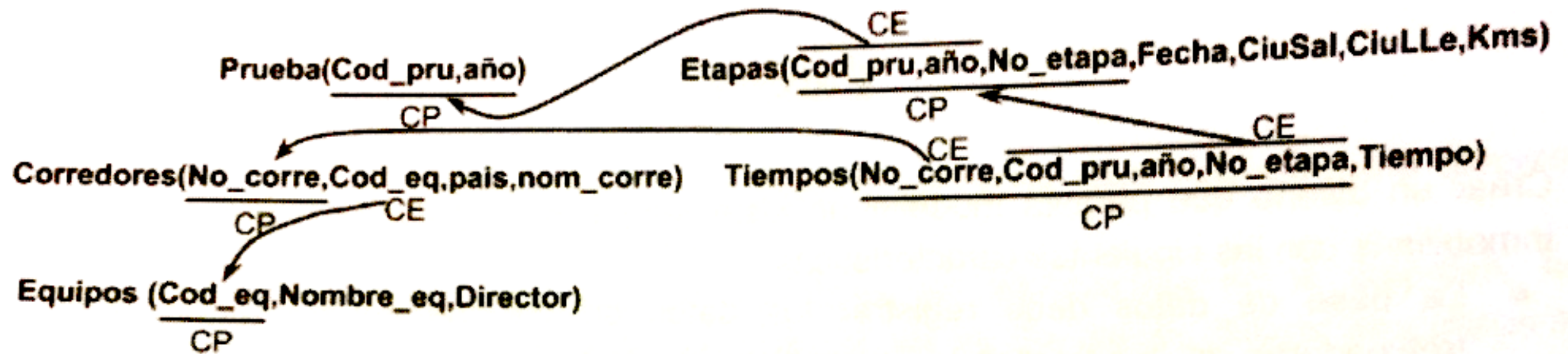



b. “Muestra el nombre de los corredores que han realizado todas las etapas de la prueba ‘Giro de Italia’ del 2015”. (en **AR 1.25pts.** y **SQL 1.25 pts.**)

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

```

SELECT nom_corre FROM Corredores WHERE No_corre IN
(SELECT DISTINCT No_corre FROM Tiempos T1 WHERE
NOT EXISTS (SELECT * FROM Etapas WHERE (Cod_pru='Giro de
Italia') AND (año=2015) AND
NOT EXISTS (SELECT * FROM Tiempos T2 WHERE
(T2.Cod_pru=Etapas.Cod_pru) AND (T2.año=Etapas.año) AND
(T2.No_etapa=Etapas.No_etapa) AND (T2.No_corre=T1.No_corre))));
  
```



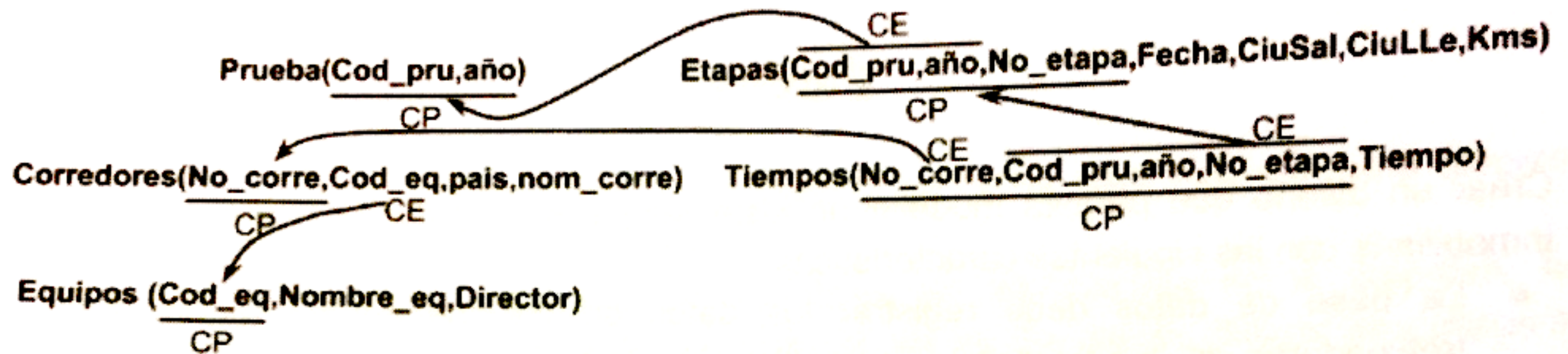
c. “Muestra los equipos cuyos corredores son todos del mismo país”. (en **CRT 1.25 pts.** y **SQL 1.25 pts.**).

$$\pi_{Cod_eq}(Equipos) - \pi_{A.Cod_eq}(\sigma_{(A.pais \neq B.pais) \wedge (A.Cod_eq = B.Cod_eq)}(\rho_A(Corredores) \times \rho_B(Corredores)))$$

```
SELECT Cod_eq FROM Corredores
GROUP BY Cod_eq
HAVING COUNT(DISTINCT pais) = 1;
```

```
(SELECT cod_eq FROM Equipo) MINUS
(SELECT X.cod_eq FROM Corredores X, CorredoresY
WHERE (X.cod_eq=Y.cod_eq) AND (X.pais!=Y.pais));
```

```
SELECT DISTINCT cod_eq FROM Corredores X WHERE NOT EXISTS (
  SELECT * FROM Corredores Y WHERE (X.cod_eq=Y.cod_eq) AND (X.pais!
    =Y.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 Vista (Nombre, TiempoTotal) AS
(SELECT nom_corre, tt FROM
  (SELECT No_corre, SUM(Tiempo) tt
   FROM Tiempos
   WHERE Cod_pru='Tour de Francia' AND año=2014
   GROUP BY No_corre) NATURAL JOIN Corredores
ORDER BY tt
);
```