APELLIDOS _____ NOMBRE _____

Instrucciones

- El examen consta de dos partes. Esta parte dura media hora. La segunda parte durará dos horas y media.
- Escribe **en esta hoja** tu nombre y la respuesta al test.
- Puedes utilizar lápiz.

PARTE 1: Test de cuestiones teórico-prácticas

1. *(2 puntos)* **SOBRE LA SIGUIENTE CUADRÍCULA, escribe para cada una de las siguientes preguntas la ÚNICA afirmación correcta (a, b, c ó d)**. Una pregunta mal contestada NO puntúa negativo.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A														
puntos según nº aciertos →			0.0	0.2	0.4	0.6	0.8	1,0	1,2	1,4	1,6	1.8	2.0	2.0

APELLIDOS _____ NOMBRE _____

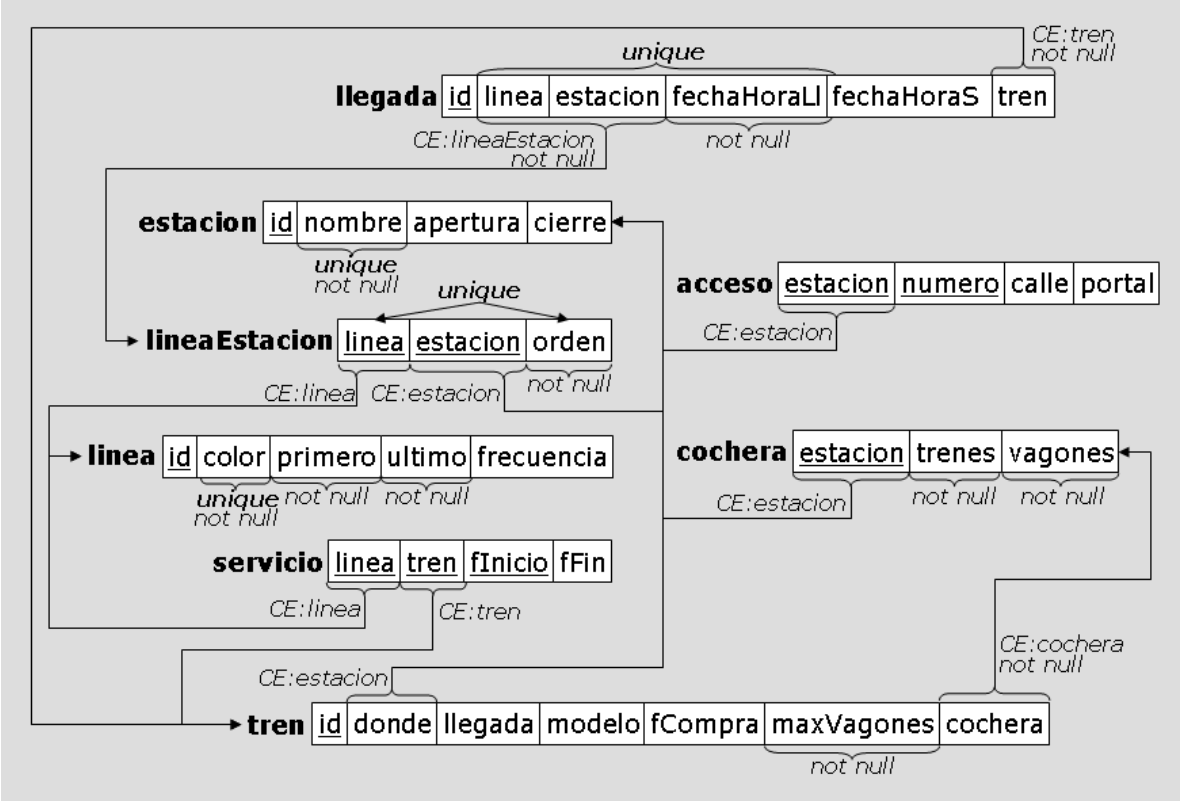
Instrucciones

- Tiempo de esta parte: 2 ½ horas. Puedes utilizar **lápiz**.
- Escribe tu nombre **en todas las hojas** que uses.
- Escribe las respuestas a los ejercicios **siguiendo el mismo orden** que en el enunciado.
- **No incluyas en las consultas ni tablas innecesarias ni *outer join* innecesarios.**
- **Recuadra cada subconsulta. Subraya y comenta cada correlación** (--correlación)**.**
- **En los join, pon el nombre de cada tabla al principio de una línea.**
- **Elimina explícitamente duplicados únicamente cuando sea necesario**

PARTE 2: Ejercicios

Se ha desarrollado un sistema informático para gestionar las ***líneas de metro de una ciudad***. La BD recoge información sobre las líneas de metro, sus estaciones y accesos desde el exterior, sus cocheras, de los trenes, su posición actual y horas de llegada y salida en las estaciones, y de las líneas en las que están o han estado en servicio los trenes.

Esquema BD (en cada tabla, el conjunto de atributos subrayados forma su clave primaria)



estacion

- Contiene el identificador numérico de la estación. Además guarda el nombre no nulo y único de cada estación y el horario de apertura y de cierre (cadenas de caracteres en formato 'hh:mm'). Si no se especifica "apertura" o "cierre" se asume el horario genérico del metro.
- Por cada estación pasa al menos una línea de metro.

linea

- En la ciudad existen varias líneas de metro. Cada una se identifica por un identificador numérico o por un color no nulo (cadena de caracteres).

- Los atributos no nulos “primero” y “ultimo” son cadenas de caracteres (formato ‘hh:mm’) que contienen el horario de partida de los primeros y últimos trenes de la línea. En cada uno de estos horarios parten dos trenes en la línea desde cada uno de sus extremos (y en dirección contraria).
- “Frecuencia” son los minutos de promedio que tarda en llegar el siguiente tren de esa línea.

lineaEstacion

- Cada línea pasa por un conjunto de estaciones. En las estaciones que pasan varias líneas, los pasajeros pueden cambiar de línea.
- Cada fila de esta tabla corresponde a una estación de una línea. Por lo tanto, en las estaciones donde pasan varias líneas, el identificador de la estación aparecerá en esta tabla tantas veces como líneas de metro pasan por ella.
- “Orden” es un número no nulo que indica el número de orden de la estación en la línea. No puede repetirse el mismo número de orden en una línea (unique).
- “Estacion” es clave extranjera que referencia a la tabla estación y “linea” otra clave extranjera que referencia a la tabla línea.

acceso

- Cada acceso desde el exterior (la calle) a una estación de metro cuenta con una fila en esta tabla. “Estación” es clave extranjera que referencia a la tabla estación y “numero” sirve para distinguir los diferentes accesos a una misma estación. Toda estación cuenta al menos con un acceso.
- Los atributos opcionales “calle” y “portal”, indican la dirección de la ciudad donde se encuentra situada la puerta de acceso. Pueden ser nulos.

tren

- Cada uno recoge información de una *máquina de tren*. “Id” es un identificador numérico.
- El sistema informático recoge periódicamente la información del GPS de cada máquina de tren y modifica los atributos “donde” y “llegada” del tren. Si el tren **está en marcha**, “donde” contiene la siguiente estación de destino (clave extranjera que referencia a la tabla estación) y “llegada” el número de minutos estimados para alcanzarla. Si el tren **está detenido** en una estación, “donde” contiene la estación donde se encuentra y “llegada” tiene el valor cero.
- Si “donde” y “llegada” tienen ***valor nulo***, significa que el tren está ubicado **en su cochera**.
- El atributo opcional “modelo” es una cadena de caracteres que contiene el modelo de la máquina de tren y el atributo opcional “fCompra” contiene la fecha de adquisición de la máquina (tipo date).
- Los atributos obligatorios “maxVagones” y “cochera” contienen el número máximo de vagones que puede arrastrar la máquina de tren y la cochera donde está asignada, respectivamente. “Cochera” es clave extranjera que referencia a la tabla cochera.

cochera

- Una cochera está situada en una estación y cada estación sólo puede tener una cochera. “Estacion” es clave extranjera que referencia a la tabla estación.
- Los atributos obligatorios “trenes” y “vagones” son sendos números que recogen el número máximo de máquinas de tren y de vagones que pueden guardarse en la cochera.

servicio

- Periodos continuados de servicio de un determinado tren a una de las líneas. Un mismo tren **puede prestar servicio a varias líneas** a la vez.
- “Linea” y “tren” son claves extranjeras que hacen referencia una a la tabla línea y la otra a tren.
- “fInicio” y “fFin” son las fechas (date) de inicio y fin del periodo de servicio del tren en la línea. **El tren presta sus servicios *actualmente* en la línea cuando “fFin” vale null.**

llegada

- Cada fila contiene datos de la llegada de un tren a un andén de una estación de una línea. Se mantienen datos de los últimos 3 meses (no se guarda fila cuando entra en una cochera)
- “Id” es un número que distingue una llegada de las demás. “Linea” y “estacion” conforman una clave extranjera que referencia a la tabla lineaEstacion. No pueden contener nulos.
- “FechaHoraLl” contiene la fecha y hora (date) no nulas de llegada del tren a la estación.
- “FechaHoraS” contiene la fecha y hora de salida del tren tras recoger y dejar pasajeros. Si contiene el valor nulo, significa que el tren todavía se encuentra detenido.

(continúa en la siguiente cara)

- "Tren" es una clave extranjera que referencia a la tabla tren y no puede contener nulos.
- No puede haber dos llegadas a la misma estación y línea con la misma hora de llegada (unique).
- Si "fechaHoraS" tiene un valor no nulo, éste será mayor que "fechaHoraLl". Ambos valores tienen bien la misma fecha, o bien "fechaHoraLl" es un día anterior a "fechaHoraS".

1. *(0,5 puntos)* Completa la siguiente **instrucción SQL** que crea la tabla **llegada**. Incluye en la instrucción los tipos y restricciones de integridad que faltan según la descripción inicial de la tabla. Se pide una instrucción y no varias.

```
CREATE TABLE llegada (
  id
  linea
  estacion
  fechaHoraLl
  fechaHoraS
  tren
   ...
  constraint llegada_salida check(   ...   )
  );
```

2. *(0,5 puntos)* Se ha construido una cochera para la estación 'Mar de Cristal' (que no tenía cochera) con capacidad para 7 máquinas de tren y 70 vagones. En esta cochera se van a derivar a todos los trenes adquiridos (fCompra) el año 2009. Escribe **las instrucciones sql** necesarias para actualizar esta información en la base de datos.

3. *(0,5 puntos)* Se desea modificar el valor 17 del atributo id de la tabla tren por el nuevo valor 27 (no existe como valor de id). Pero el tren 17 aparece tanto en filas de la tabla llegada como en filas de la tabla servicio. Escribe las **instrucciones sql** necesarias para hacer estos cambios sin insertar ni eliminar filas en las tablas de la BD. Ten en cuenta que la restricción de clave primaria de la tabla tren se llama pk_tren y que las restricciones de clave extranjera que referencia a la tabla tren de las tablas llegada y servicio se llaman ambas fk_tren. Ambas claves extranjeras han sido definidas con la opción *on update restrict.*

4. *(0,5 puntos)* Se desea modificar la tabla estación de manera que las columnas llamadas apertura y cierre pasen a llamarse horaApertura y horaCierre. Además debemos garantizar que, o bien las horas de apertura y cierre son nulas, o bien ambas son no nulas y la de apertura es menor que la de cierre (lee en la descripción de la tabla cuál es el tipo de datos y formato de ambos atributos). Escribe las **instrucciones sql** necesarias para hacer estos cambios en la definición de la tabla.

5. *(0,5 puntos)* Escribe una **consulta sql** que obtenga para cada tren fuera de servicio *(que no esté en servicio de ninguna línea. Lee lo que se explica en la tabla servicio sobre el valor de "fFin")* su identificador, modelo, fecha de compra y nombre de la estación en la que está su cochera.

6. *(0,5 puntos)* Crea una **vista** llamada **lineasAccesos** que devuelva el identificador y nombre de cada una de las estaciones junto al número de líneas de metro que pasan por la estación y el número de accesos que tiene la estación desde la calle.

7. *(0,5 puntos)* Crea una **vista** llamada **lineasTrenesCochera** que devuelva el nombre de cada estación con algún tren en su cochera, junto al número de líneas de metro que pasan por la estación y el número de trenes estacionados en su cochera. Nos interesa únicamente una solución que haga uso de la vista **lineasAccesos**.

8. *(1 punto)* Escribe una **consulta sql** que obtenga para cada tren su identificador, el nombre de la estación donde o bien se encuentra actualmente, o bien está llegando, o bien está estacionado en su cochera, y el número de veces que se ha detenido en alguna estación durante la jornada actual (sysdate) según figura en la tabla "llegada". Puede haber trenes que hayan permanecido parados toda la jornada, pero también queremos que aparezcan en la solución. Las tres columnas del resultado deberán llamarse: tren, estación y llegadas.

9. *(1 punto)* Escribe una **consulta sql** que obtenga el identificador, modelo y fecha de compra (sin repeticiones) de aquellos trenes que estén prestando ***actualmente*** servicio a la línea azul (según la tabla servicio) y que hayan permanecido algún día completo en la cochera durante los últimos siete días (desde sysdate-7). Esto es, los trenes para los que no figura actividad en la tabla llegada alguno de los días del rango mencionado.

10. *(1 punto)* Escribe una **consulta sql** que obtenga el identificador de cada estación y línea que pase por ella, junto a la suma de los tiempos de espera al siguiente tren *(el que transcurre entre que sale un tren de una línea de una estación y llega el siguiente)* respecto a las llegadas a estación durante **el día de hoy (sysdate)**. No confundir el tiempo de espera que se menciona con el tiempo que permanece detenido un tren en una estación ni tampoco con el atributo frecuencia de la tabla línea (que es el promedio estipulado).

    *Para calcular los **minutos** de diferencia (en valor absoluto) entre 2 valores de tipo date podemos disponer de la función: **minutos_entre(**date1**,** date2**)**.*

11. *(1 punto)* Escribe una **consulta sql** que obtenga el identificador, modelo y fecha de compra (id, modelo y fCompra) de las máquinas de tren compradas los últimos 3 años que hayan prestado servicio (o lo estén prestando en la actualidad) en **todas** las líneas con alguna estación accesible desde la calle "Serrano".

12. *(0,5 puntos)* Escribe una **consulta en álgebra relacional** que muestre los nombres de las estaciones con cochera que tienen su cochera vacía (sin trenes). Ten en cuenta que en una cochera sólo pueden estar trenes que tengan asignada esa cochera (según el atributo cochera de la tabla tren). Se sabe si un tren está en la cochera según el valor de su atributo "donde".

ALGUNAS FUNCIONES DE ORACLE (puede que no hagan falta todas)

Nombre(argumentos)	Significado	Ejemplos
sysdate	Devuelve la fecha actual	**sysdate** → '16-jun-2009'
to_char(<fecha>, <formato>)	Convierte la fecha a cadena según el formato especificado	**to_char(sysdate,'YYYY')** → '2009' **to_char(sysdate,'YYYY/MM/DD')** → '2009/06/16'
to_date(<cadena>,<formato>)	Convierte la cadena dada según el formato especificado a fecha	**to_date('16-06-09','DD-MM-YY')** → [fecha de hoy]
substr(<cadena>,<inicio>,<avance>)	Extrae una subcadena de <cadena> de longitud <avance> empezando en el carácter <inicio>.	**substr('mi cadena', 5, 3)** → 'ade'

2. *(0,5 puntos)*

```
CREATE TABLE llegada (
id          int not null,
linea       int not null,
estacion    int not null,
fechaHoraLl date not null,
fechaHoraS  date,
tren        int not null,

constraint pk_llegada
            primary key (id),
constraint fk_lineaEstacion
            foreign key (linea, estacion)
            references lineaEstacion(linea, estacion),
constraint fk_tren
            foreign key (tren)
            references tren(id),
constraint u_linEstLleg
            unique(linea, estacion, fechaHoraLl),
constraint llegada_salida
            check( fechaHoraS is null or
                   (fechaHoraLl<fechaHoraS and
                     to_char(fechaHoraLl,'dd-mm-yyyy')+1>=
                     to_char(fechaHoraS, 'dd-mm-yyyy')
                    ))
);
```

3. *(0,5 puntos)* Se ha construido una cochera para la estación 'Mar de Cristal' (que no tenía cochera) con capacidad para 7 máquinas de tren y 70 vagones. En esta cochera se van a derivar a todos los trenes adquiridos (fCompra) el año 2009. Escribe **las instrucciones sql** necesarias para actualizar esta información en la base de datos.

insert into cochera
  values ((select id from estacion where nombre='Mar de cristal'), 7, 70);
update tren
  set cochera=(select id from estacion where nombre='Mar de cristal')
  where to_char(fCompra,'yyyy')=2009;

4. *(0,5 puntos)* Se desea modificar el valor 17 del atributo id de la tabla tren por el nuevo valor 27 (no existe como valor de id). Pero el tren 17 aparece tanto en filas de la tabla llegada como en filas de la tabla servicio. Escribe las **instrucciones sql** necesarias para hacer estos cambios sin insertar ni eliminar filas en las tablas de la BD. Ten en cuenta que la restricción de clave primaria de la tabla tren se llama pk_tren y que las restricciones de clave extranjera que referencia a la tabla tren de las tablas llegada y servicio se llaman ambas fk_tren. Ambas claves extranjeras han sido definidas con la opción *on update restrict.*

***alter table** llegada **disable constraint** fk_tren;*
***alter table** servicio **disable constraint** fk_tren;*
***update** tren **set** id=27 **where** id=17;*
***update** llegada **set** tren=27 **where** tren=17;*
***update** servicio **set** tren=27 **where** tren=17;*
***alter table** llegada **enable constraint** fk_tren;*
***alter table** servicio **enable constraint** fk_tren;*

5. *(0,5 puntos)* Se desea modificar la tabla estación de manera que las columnas llamadas apertura y cierre pasen a llamarse horaApertura y horaCierre. Además debemos garantizar que, bien las horas de apertura y cierre son nulas, o bien ambas son no nulas y la de apertura es menor que la de cierre (lee en la descripción de la tabla cuál es el tipo de datos y formato de ambos atributos). Escribe las **instrucciones sql** necesarias para hacer estos cambios en la definición de la tabla.

```
alter table estacion rename column apertura to horaApertura;
alter table estacion rename column cierre    to horaCierre;
alter table estacion add constraint ordenHoras
            check ((horaApertura is null and horaCierre is null) or horaApertura<horaCierre)
```

6. *(0,5 puntos)* Escribe una **consulta sql** que obtenga para cada tren fuera de servicio *(que no esté en servicio de ninguna línea. Lee lo que se explica en la tabla servicio sobre el valor de "fFin")* su identificador, modelo, fecha de compra y nombre de la estación en la que está su cochera.

```
select  T.id, T.modelo, T.fCompra, E.nombre
from  estacion E join
        tren T on E.id=T.cochera
except
select T.id, T.modelo, T.fCompra, E.nombre
from  estacion E join
        tren T on E.id=T.cochera join
        servicio S on T.id=S.tren
where S.fFin is null
```

Otra forma:

```
select  T.id, T.modelo, T.fCompra, E.nombre
from  estacion E join
        tren T on E.id=T.cochera
where T.id not in (select tren from servicio where fFin is null)
```

7. *(0,5 puntos)* Crea una **vista** llamada **lineasAccesos** que devuelva el identificador y nombre de cada una de las estaciones junto al número de líneas de metro que pasan por la estación y el número de accesos que tiene la estación desde la calle.

```
create view lineasAccesos as
   select  E.id, E.nombre, count(distinct E.linea) as numLineas,
             count(distinct A.numero) as numAccesos
   from  estacion E join
           lineaEstacion LE on E.id=LE.estacion join
           acceso A on E.id=A.estacion
   group by E.id, E.nombre
```

8. *(0,5 puntos)* Crea una **vista** llamada **lineasTrenesCochera** que devuelva el nombre de cada estación con algún tren en su cochera, junto al número de líneas de metro que pasan por la estación y el número de trenes estacionados en su cochera. Nos interesa únicamente una solución que haga uso de la vista **lineasAccesos**.

```
create view lineasTrenesCochera as
   select  LA.nombre, LA.numLineas, count(T.id)
   from  lineasAccesos LA join
           tren T on LA.id=T.cochera
   where T.donde is null
   group by LA.nombre, LA.numLineas
```

9. *(1 punto)* Escribe una **consulta sql** que obtenga para cada tren su identificador, el nombre de la estación donde o bien se encuentra actualmente, o bien está llegando, o bien está estacionado en su cochera, y el número de veces que se ha detenido en alguna estación durante la jornada actual (sysdate) según figura en la tabla "llegada". Puede haber trenes que hayan permanecido parados toda la jornada, pero también queremos que aparezcan en la solución. Las tres columnas del resultado deberán llamarse: tren, estación y llegadas.

```
select  T.id as tren, coalesce(E.nombre, C.nombre) as estacion, count(L.id) as llegadas
from  tren T left outer join
        estacion E on T.donde=E.id join
        estacion C on T.cochera=C.id left outer join
        llegada L on T.id=L.tren and L.fecha=sysdate
group by  T.id, E.nombre, C.nombre
```

10. *(1 punto)* Escribe una **consulta sql** que obtenga el identificador, modelo y fecha de compra (sin repeticiones) de aquellos trenes que estén prestando *actualmente* servicio a la línea azul (según la tabla servicio) y que hayan permanecido algún día completo en la cochera durante los últimos siete días (desde sysdate-7). Esto es, los trenes para los que no figura actividad en la tabla llegada alguno de los días del rango mencionado.

```
select  T.id, T.modelo, T.fCompra
from  tren T join
        servicio S on T.id=S.tren join
        linea L on L.id=S.linea
where  L.color='azul' and S.fFin is null and 7<>
        (select count(distinct to_char(L.fechaHoraLl,'dd-mm-yyyy'))
         from llegada L
         where L.tren=T.id --correlacion
                 and L.fechaHoraLl>=sysdate-7
        )
```

11. *(1 punto)* Escribe una **consulta sql** que obtenga el identificador de cada estación y línea que pase por ella, junto a la suma de los tiempos de espera al siguiente tren *(el que transcurre entre que sale un tren de una línea de una estación y llega el siguiente)* respecto a las llegadas a estación durante **el día de hoy (sysdate)**. No confundir el tiempo de espera que se menciona con el tiempo que permanece detenido un tren en una estación, ni tampoco con el atributo frecuencia de la tabla línea (que es el promedio estipulado).

    *Para calcular los **minutos** de diferencia (en valor absoluto) entre 2 valores de tipo date podemos disponer de la función: **minutos_entre(**date1**,** date2**)**.*

```
select  LE.estacion, LE.linea,
        sum(minutos_entre(L1.fechaHoraS, L2.fechaHoraLl)) as “espera acumulada”
from  lineaEstacion LE left outer join
      llegada L1 on LE.linea=L1.linea and LE.estacion=L1.estacion join
      llegada L2 on LE.linea=L2.linea and LE.estacion=L2.estacion
where  L1.fechaHoraLl=sysdate and L2.fechaHoraLl=sysdate and
       L1.fechaHoraS=
    (select max(LL.fechaHoraS)
    from llegada LL
    where LL.fechaHoraLl=sysdate and
          LL.estacion=LE.estacion and --correlacion
          LL.linea=LE.linea and --correlacion
          LL.fechaHoraS<L2.fechaHoraLl --correlacion
    )
group by  LE.estacion, LE.linea
```

12. *(1 punto)* Escribe una **consulta sql** que obtenga el identificador, modelo y fecha de compra (id, modelo y fCompra) de las máquinas de tren compradas los últimos 3 años que hayan prestado servicio en **todas** las líneas con alguna estación accesible desde la calle “Serrano”.

```
select  T.id, T.modelo, T.fCompra
from  tren T
where extract(year from sysdate)-3 <= extract(year from T.fCompra) and
      not exists
    ( select LE.linea
      from lineaEstacion LE join
           acceso A on LE.estacion=A.estacion
      where A.calle=’Serrano’
      except
      select S.linea
      from servicio S
      where S.tren=T.id --correlacion
    )
```

13. *(0,5 puntos)* Escribe una **consulta en álgebra relacional** que muestre los nombres de las estaciones con cochera que tienen su cochera vacía (sin trenes). Ten en cuenta que en una cochera sólo pueden estar trenes que tengan asignada esa cochera (según el atributo cochera de la tabla tren). Se sabe si un tren está en la cochera según el valor de su atributo "donde".

En sql se escribiría, por ejemplo, así:

```
select  E.nombre
from  estacion E join
        cochera C on E.id=C.estacion
except
select E.nombre
from estacion E join
       tren T on E.id=T.cochera
where T.donde is null
```

Nos piden su escritura en álgebra relacional:

EstConCoch ← π_{nombre} (estacion $|X|_{id=estacion}$ cochera)
TrenEnCoch ← $\sigma_{donde=null}$ tren
EstTrenEnCoch ← π_{nombre} (TrenEnCoch $|X|_{cochera=id}$ estacion)
Resultado ← EstConCoch − EstTrenEnCoch