

Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Bases de datos

Segundo Cuatrimestre del 2011

Trabajo Práctico 1: MER, MR, SQL

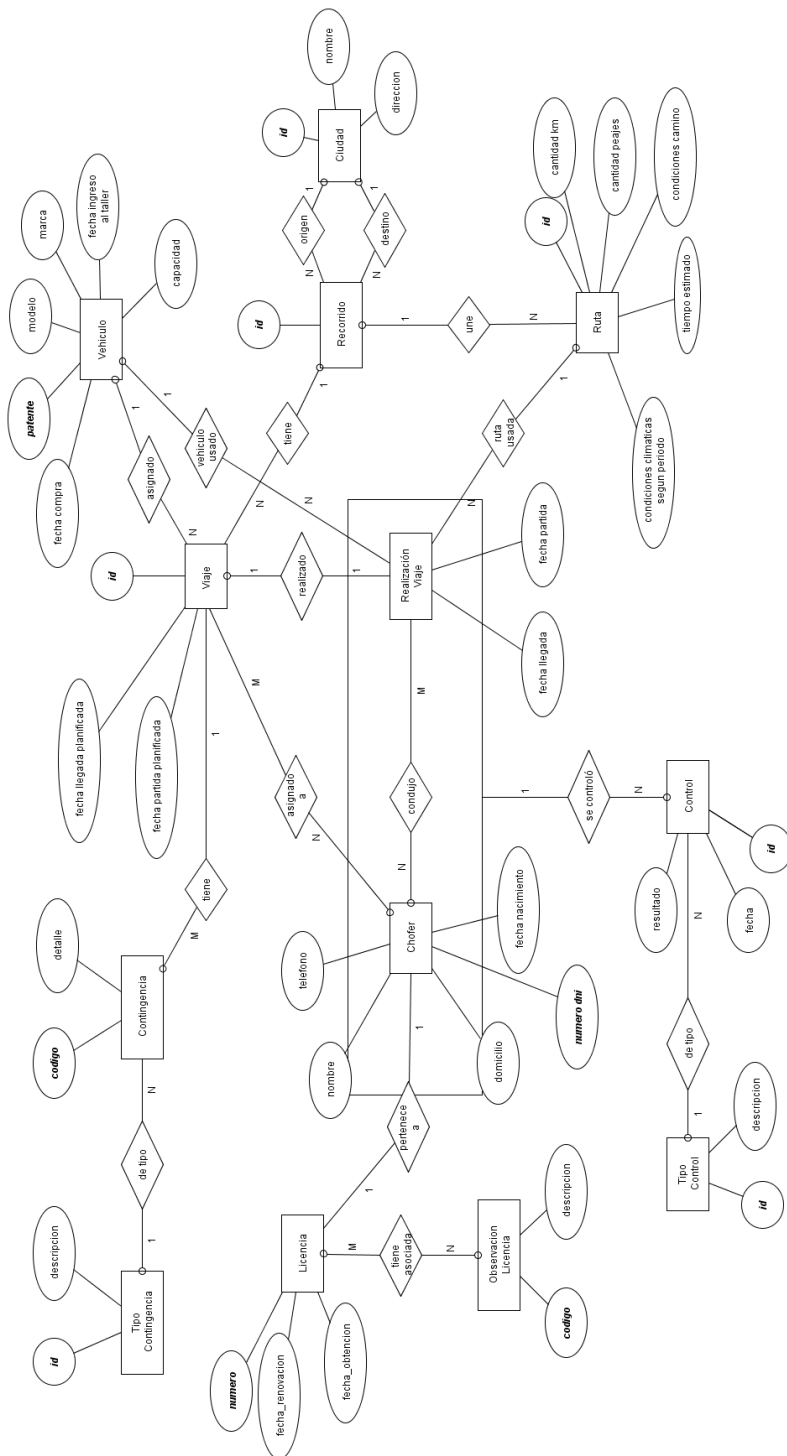
Integrante	LU	Correo electrónico
Ezequiel Castellano	161/08	ezequielcastellano@gmail.com
Juan Edi	?/08	jedi11235@gmail.com
Carolina Hadad	367/08	carolinahadad@gmail.com
Mariano Semelman	?/08	marianosemelman@gmail.com

ÍNDICE	2
--------	---

Índice

1. MER	3
1.1. Decisiones tomadas	4
2. MR	4
2.1. Esquema	4
3. SQL	6
3.1. Stored Procedures	6
3.2. Triggers	8
3.3. Diseño Físico	9

1. MER



1.1. Decisiones tomadas

1. Modelamos las observaciones como una entidad separada, por que en el dominio podría ser importante encontrar todos los choferes que usen anteojos, por ejemplo. Las observaciones se discriminan por estar tipificadas.
2. La antigüedad del vehículo, se deriva a partir de su fecha de compra.
3. Se modela que no hay peajes poniendo un 0 y no consideramos que sea una entidad separada, pues no hay otra información relevante sobre ellos.
4. En un principio habíamos modelado el campo condiciones climáticas pensándolo como una relación con un atributo asociado según el periodo del año/condición climática, sin embargo descartamos esta solución pues nos hacia modelar cosas por demás que no encontramos necesario según en el enunciado.
5. Modelamos los viajes realizados asociándolos a un viaje, para no perder la información de un viaje planificado al realizarlo y mantener la relación entre estos.
6. Como constraint la aridad de la relación chofer asignado a viaje planificado es como máximo tres choferes asignados a un viaje planificado.
7. En los vehículos, asumimos que el estado es si está en reparación o no, y no algo adicional. Un vehículo se encuentra en reparación si tiene fecha de ingreso al taller. No guardamos el historial de reparaciones realizadas.
8. En nuestro modelo los viajes realizados puede haber sido hechos por choferes con licencias vencidas. Si modeláramos esto nos limitaría muchas situaciones posibles validas. Además como no guardamos historial de licencias, no podemos saber si hizo un viaje con la licencia vencida.
9. La planificación de los viajes se realiza en función de un solo vehículo.
10. Tanto el vehículo como el chofer de un viaje realizado pueden ser diferentes a los propuestos durante la planificación.
11. Para guardar la información sobre los viajes planificados una vez que ser realizan, los modelamos como tablas distintas. La PK del viaje realizado es la FK al viaje planificado.

2. MR

2.1. Esquema

TipoContingencia(id, descripción)

PK = CK = {idTipoContingencia}

Contingencia(código, descripción, idTipoContingencia, idViaje)

PK = CK = {código}

FK = {idTipoContingencia, idViaje}

Viaje(idViaje, fechaPartidaPlanificada, fechaLlegadaPlanificada, idRecorrido, idVehiculo)

PK = CK = {idViaje}

FK = { idRecorrido, idVehiculo}

RealizacionViaje(idViaje, fechaPartida, fechaLlegada, idRuta, idVehiculo)

PK = CK = {idViaje}

FK = { idViaje, idRuta, idVehiculo}

Vehiculo(patente, fechaDeCompra, modelo, marca, capacidad, fechaDeIngresoAlTaller)

PK = CK = {patente}

FK = {}

Recorrido(idRecorrido, idCiudadOrigen, idCiudadDestino)

PK = CK = {idRecorrido}

FK = {idCiudadOrigen, idCiudadDestino}

Ciudad(idCiudad, nombre, dirección)

PK = {idCiudad}

CK = {direccion}

FK = {}

Ruta(idRuta, cantidadKm, cantidadPeajes, condicionesCamino, tiempoEstimado, condicionesClimaticasSegunPeriodo)

PK = CK = {idRuta}

FK = {}

Control(idControl, resultado, fecha, idTipoControl, idChofer, idViaje)

PK = CK = {idControl}

FK = {(idChofer, idViaje), idTipoControl}

TipoControl(idTipoControl, descripcion)

PK = CK = {idTipoControl}

FK = {}

Licencia(numero, fechaRenovacion, fechaObtencion)

PK = CK = {numero}

FK = {}

ObservacionLicencia(codigo, descripcion,)

PK = CK = {codigo}

FK = {}

TieneAsociada(idLicencia, idObservacionLicencia)

PK = CK = {(idLicencia, idObservacionLicencia)}

FK = {idLicencia, idObservacionLicencia}

Chofer(codigo, descripcion, idLicencia)

PK = CK = {codigo}

FK = {idLicencia}

AsignadoA(idChofer, idViaje)

PK = CK = {(idLicencia, idObservacionLicencia)}

FK = {idLicencia, idObservacionLicencia}

Conduce(idChofer, idViajeRealizado)

PK = CK = {(idChofer, idViajeRealizado)}

FK = {idChofer, idViajeRealizado}

3. SQL

3.1. Stored Procedures

```
ï»¿DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS 'recorridosConTodasRutasUsadas' $$
```

```

CREATE PROCEDURE 'recorridosConTodasRutasUsadas'()
BEGIN

SELECT re.* FROM recorrido re
LEFT JOIN ruta ru
      LEFT JOIN realizacion_viaje rv
      ON ru.id = rv.ruta
ON ru.recorrido=re.id
WHERE DATE_SUB(CURDATE(),INTERVAL 1 YEAR) < rv.fecha_partida
GROUP BY re.id
HAVING count(ru.id)>1 and count(ru.id) = count(rv.ruta);

END $$

DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS 'choferesQueManejaronTodosLosNuevos' $$
CREATE PROCEDURE 'choferesQueManejaronTodosLosNuevos'()
BEGIN

DECLARE cantidad INT(8) DEFAULT 0;
SELECT count(*) INTO cantidad FROM vehiculo WHERE fecha_compra > DATE_SUB(CURDATE(),INTERVAL 1 YEAR)
SELECT c.* FROM chofer c
LEFT JOIN realizacion_viaje_chofer rvc
      LEFT JOIN realizacion_viaje rv
      ON rv.viaje = rvc.realizacion_viaje
ON rvc.chofer=c.numero_dni
GROUP BY c.numero_dni
HAVING count(distinct rv.vehiculo) = cantidad;

END $$

DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS 'viajesRealizadosYEstadoPorVehiculo' $$
CREATE PROCEDURE 'viajesRealizadosYEstadoPorVehiculo'()
BEGIN

SELECT v.patente patente ,
      count(r.viaje) / (1 + YEAR(CURRENT_DATE()) - YEAR(fecha_compra)) viajesPromedio ,
      IF(fecha_ingreso_taller is null , false , true) enReparacion
FROM vehiculo v
LEFT JOIN realizacion_viaje r
      ON r.vehiculo = v.patente
GROUP BY v.patente;

```

```
END $$
```

```
DELIMITER ;
```

3.2. Triggers

```
DELIMITER $$
```

```
CREATE TRIGGER evitarPlanificarConCochesEnReparacion BEFORE INSERT ON viaje  
FOR EACH ROW BEGIN
```

```
    DECLARE enUso BOOLEAN;
```

```
    SELECT (fecha_ingreso_taller is null) INTO enUso FROM vehiculo v WHERE v.patente=NEW.ve
```

```
    IF NOT enUso THEN
```

```
        SET NEW.vehiculo=null;
```

```
    END IF;
```

```
END $$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE TRIGGER evitarRealizarViajesConCochesEnReparacion BEFORE INSERT ON realizacion_viaje  
FOR EACH ROW BEGIN
```

```
    DECLARE enUso BOOLEAN;
```

```
    SELECT (fecha_ingreso_taller is null) INTO enUso FROM vehiculo v WHERE v.patente=NEW.ve
```

```
    IF NOT enUso THEN
```

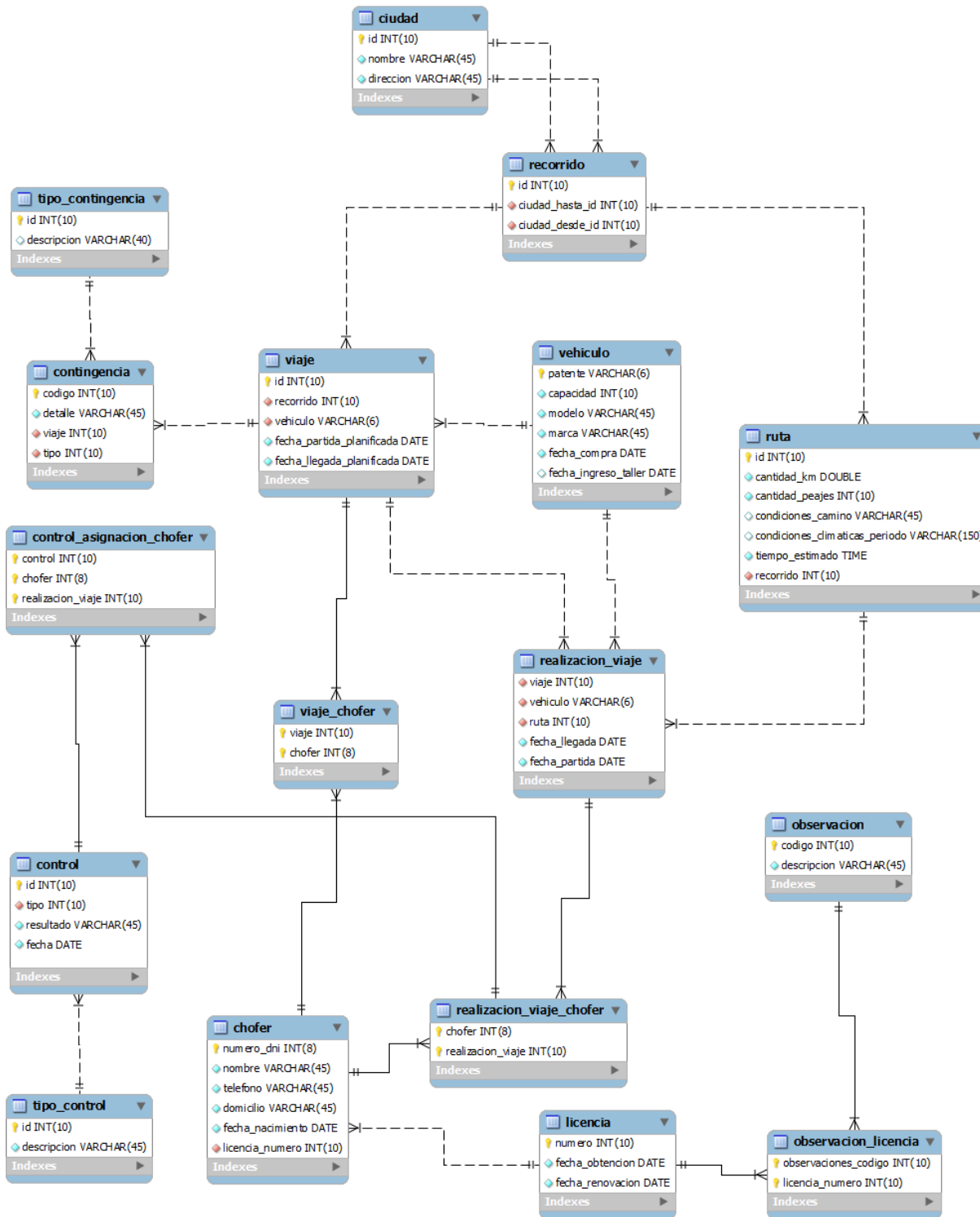
```
        SET NEW.vehiculo=null;
```

```
    END IF;
```

```
END $$
```

```
DELIMITER ;
```


3.3. Diseño Físico



```

CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE 'chofer' (
    'numero_dni' int(8) unsigned NOT NULL,
    'nombre' varchar(45) NOT NULL,
    'telefono' varchar(45) NOT NULL,
    'domicilio' varchar(45) NOT NULL,
    'fecha_nacimiento' date NOT NULL,
    'licencia_numero' int(10) unsigned NOT NULL,
    PRIMARY KEY ('numero_dni') USING BTREE,
    UNIQUE KEY 'chofer_licencia_numero' ('licencia_numero') USING BTREE,
    CONSTRAINT 'chofer_licencia_numero' FOREIGN KEY ('licencia_numero') REFERENCES 'licencia' (
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE 'ciudad' (
    'id' int(10) unsigned NOT NULL AUTO_INCREMENT,
    'nombre' varchar(45) NOT NULL,
    'direccion' varchar(45) NOT NULL,
    PRIMARY KEY ('id'),
    UNIQUE KEY 'nombre_direccion' ('nombre','direccion')
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

CREATE TABLE 'contingencia' (
    'codigo' int(10) unsigned NOT NULL AUTO_INCREMENT,
    'detalle' varchar(45) NOT NULL,
    'viaje' int(10) unsigned NOT NULL,
    'tipo' int(10) unsigned NOT NULL,
    PRIMARY KEY ('codigo'),
    KEY 'contingencia_tipo_fk' ('tipo'),
    KEY 'contingencia_viaje_fk' ('viaje'),
    CONSTRAINT 'contingencia_tipo_fk' FOREIGN KEY ('tipo') REFERENCES 'tipo_contingencia' ('id'),
    CONSTRAINT 'contingencia_viaje_fk' FOREIGN KEY ('viaje') REFERENCES 'viaje' ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE 'control' (
    'id' int(10) unsigned NOT NULL AUTO_INCREMENT,
    'tipo' int(10) unsigned NOT NULL,
    'resultado' varchar(45) NOT NULL,
    'fecha' date NOT NULL,
    PRIMARY KEY ('id') USING BTREE,
    KEY 'control_tipo_control_fk' ('tipo'),
    CONSTRAINT 'control_tipo_control_fk' FOREIGN KEY ('tipo') REFERENCES 'tipo_control' ('id')
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

CREATE TABLE 'control_asignacion_chofer' (
    'control' int(10) unsigned NOT NULL,
    'chofer' int(8) unsigned NOT NULL,
    'realizacion_viaje' int(10) unsigned NOT NULL,
    PRIMARY KEY ('chofer','control','realizacion_viaje'),
    KEY 'control_asignacion_chofer_chofer_fk' ('chofer'),

```

```

KEY 'control_asignacion_chofer_chofer_realizacion_viaje_fk' ('chofer','realizacion_viaje'),
KEY 'control_asignacion_chofer_control_fk' ('control'),
CONSTRAINT 'control_asignacion_chofer_chofer_realizacion_viaje_fk' FOREIGN KEY ('chofer',
CONSTRAINT 'control_asignacion_chofer_control_fk' FOREIGN KEY ('control') REFERENCES 'con
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE 'licencia' (
  'numero' int(10) unsigned NOT NULL,
  'fecha_obtencion' date NOT NULL,
  'fecha_renovacion' date NOT NULL,
  PRIMARY KEY ('numero')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE 'observacion' (
  'codigo' int(10) unsigned NOT NULL AUTO_INCREMENT,
  'descripcion' varchar(45) NOT NULL,
  PRIMARY KEY ('codigo')
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

```

```

CREATE TABLE 'observacion_licencia' (
  'observaciones_codigo' int(10) unsigned NOT NULL,
  'licencia_numero' int(10) unsigned NOT NULL,
  PRIMARY KEY ('observaciones_codigo','licencia_numero') USING BTREE,
  KEY 'observaciones_licencia_numero' ('licencia_numero'),
  CONSTRAINT 'observaciones_licencia_codigo' FOREIGN KEY ('observaciones_codigo') REFERENCES
  CONSTRAINT 'observaciones_licencia_numero' FOREIGN KEY ('licencia_numero') REFERENCES 'li
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE 'realizacion_viaje' (
  'viaje' int(10) unsigned NOT NULL,
  'vehiculo' varchar(6) NOT NULL,
  'ruta' int(10) unsigned NOT NULL,
  'fecha_llegada' date NOT NULL,
  'fecha_partida' date NOT NULL,
  KEY 'realizacion_viaje_viaje_fk' ('viaje'),
  KEY 'realizacion_viaje_vehiculo_fk' ('vehiculo'),
  KEY 'realizacion_viaje_ruta_fk' ('ruta'),
  CONSTRAINT 'realizacion_viaje_ruta_fk' FOREIGN KEY ('ruta') REFERENCES 'ruta' ('id') ON D
  CONSTRAINT 'realizacion_viaje_vehiculo_fk' FOREIGN KEY ('vehiculo') REFERENCES 'vehiculo'
  CONSTRAINT 'realizacion_viaje_viaje_fk' FOREIGN KEY ('viaje') REFERENCES 'viaje' ('id') O
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE 'realizacion_viaje_chofer' (
  'chofer' int(8) unsigned NOT NULL,
  'realizacion_viaje' int(10) unsigned NOT NULL,
  PRIMARY KEY ('chofer','realizacion_viaje'),
  KEY 'realizacion_viaje_chofer_chofer_fk' ('chofer'),
  KEY 'realizacion_viaje_chofer_viaje' ('realizacion_viaje'),
  CONSTRAINT 'realizacion_viaje_chofer_chofer_fk' FOREIGN KEY ('chofer') REFERENCES 'chofer'
  CONSTRAINT 'realizacion_viaje_chofer_realizacion_viaje_fk' FOREIGN KEY ('realizacion_viaj

```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE 'recorrido' (
  'id' int(10) unsigned NOT NULL AUTO_INCREMENT,
  'ciudad_hasta_id' int(10) unsigned NOT NULL,
  'ciudad_desde_id' int(10) unsigned NOT NULL,
  PRIMARY KEY ('id'),
  UNIQUE KEY 'recorridos_unicos' ('ciudad_hasta_id', 'ciudad_desde_id'),
  KEY 'recorrido_ciudad_desde_fk' ('ciudad_desde_id'),
  KEY 'recorrido_ciudad_hasta_fk' ('ciudad_hasta_id'),
  CONSTRAINT 'recorrido_ciudad_desde_fk' FOREIGN KEY ('ciudad_desde_id') REFERENCES 'ciudad' ('id'),
  CONSTRAINT 'recorrido_ciudad_hasta_fk' FOREIGN KEY ('ciudad_hasta_id') REFERENCES 'ciudad' ('id'),
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE 'ruta' (
  'id' int(10) unsigned NOT NULL AUTO_INCREMENT,
  'cantidad_km' double NOT NULL,
  'cantidad_peajes' int(10) unsigned NOT NULL,
  'condiciones_camino' varchar(45) DEFAULT NULL,
  'condiciones_climaticas_periodo' varchar(150) DEFAULT NULL,
  'tiempo_estimado' time NOT NULL,
  'recorrido' int(10) unsigned NOT NULL,
  PRIMARY KEY ('id') USING BTREE,
  KEY 'ruta_recorrido_fk' ('recorrido'),
  CONSTRAINT 'ruta_recorrido_fk' FOREIGN KEY ('recorrido') REFERENCES 'recorrido' ('id'),
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE 'tipo_contingencia' (
  'id' int(10) unsigned NOT NULL AUTO_INCREMENT,
  'descripcion' varchar(40) DEFAULT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE 'tipo_control' (
  'id' int(10) unsigned NOT NULL,
  'descripcion' varchar(45) NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE 'vehiculo' (
  'patente' varchar(6) NOT NULL,
  'capacidad' int(10) unsigned NOT NULL,
  'modelo' varchar(45) NOT NULL,
  'marca' varchar(45) NOT NULL,
  'fecha_compra' date NOT NULL,
  'fecha_ingreso_taller' date DEFAULT NULL,
  PRIMARY KEY ('patente') USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE 'viaje' (
```

```

'id' int(10) unsigned NOT NULL AUTO_INCREMENT,
'recorrido' int(10) unsigned NOT NULL,
'vehiculo' varchar(6) NOT NULL,
'fecha_partida_planificada' date NOT NULL,
'fecha_llegada_planificada' date NOT NULL,
PRIMARY KEY ('id'),
KEY 'viaje_recorrido_id' ('recorrido'),
KEY 'viaje_recorrido_fk' ('recorrido'),
KEY 'viaje_vehiculo_fk' ('vehiculo'),
CONSTRAINT 'viaje_recorrido_fk' FOREIGN KEY ('recorrido') REFERENCES 'recorrido' ('id') ON DELETE CASCADE,
CONSTRAINT 'viaje_vehiculo_fk' FOREIGN KEY ('vehiculo') REFERENCES 'vehiculo' ('patente') ON DELETE CASCADE,
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

CREATE TABLE 'viaje_chofer' (
'viaje' int(10) unsigned NOT NULL,
'chofer' int(8) unsigned NOT NULL,
PRIMARY KEY ('viaje','chofer'),
KEY 'viaje_chofer_viaje_fk' ('viaje'),
KEY 'viaje_chofer_chofer_fk' ('chofer'),
CONSTRAINT 'viaje_chofer_chofer_fk' FOREIGN KEY ('chofer') REFERENCES 'chofer' ('numero_documento') ON DELETE CASCADE,
CONSTRAINT 'viaje_chofer_viaje_fk' FOREIGN KEY ('viaje') REFERENCES 'viaje' ('id') ON DELETE CASCADE,
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```