**Álgebra relacional**:

DUEÑO (id\_dueño, nombre, teléfono, dirección, dni)

CHOFER (id\_chofer, nombre, tel, dir, f\_licencia\_desde, f\_licencia\_hasta, dni)

AUTO (patente, id\_dueño, id\_chofer, marca, modelo, año)

- Hallar el DNI, nombre y teléfono de todos los dueños -que no sean choferes- y que solamente poseen vehículos marca "FIAT".

D\_CHOFER <- π (id\_dueño, nombre, teléfono, dirección, dni) DUEÑO |X| CHOFER

D\_NO\_CHOFER <- D\_CHOFER - DUEÑO

FIAT <- σ (marca = “FIAT”) (AUTO)

DISTINTO\_A\_FIAT <- σ (marca != “FIAT”) (AUTO)

D\_FIAT <- D\_NO\_CHOFER |X| FIAT

D\_OTROS <- D\_NO\_CHOFER |X| DISTINTO\_A\_FIAT

D\_SOLO\_FIAT <- D\_FIAT - D\_OTROS

OJO, no hacer con AUTO directamente porque voy a terminan borrando todo.

**MySQL**:

DUEÑO (id\_dueño, nombre, teléfono, dirección, dni)

CHOFER (id\_chofer, nombre, tel, dir, f\_licencia\_desde, f\_licencia\_hasta, dni)

AUTO (patente, id\_dueño, id\_chofer, marca, modelo, año)

- Utilizando el esquema presentado en el ejercicio de Álgebra Relacional, escriba un Store Procedure en MySQL llamado "nuevo\_chofer". Este debe declarar como parámetro un dni y 2 fechas (licencia\_desde y licencia\_hasta), y debe generar, dentro de una transacción, un nuevo chofer con los datos recuperados del dueño con dni igual al recibido como parámetro. Asuma que este dni ya existe en DUEÑO (esquema del ej .de AR).

DELIMITER //

CREATE PROCEDURE agregarAppointment

(IN new\_dni INT, IN f\_l\_d DATE, IN f\_l\_h DATE)

BEGIN

START TRANSACTION;

SELECT nombre, tel, dir

INTO new\_nombre, new\_tel, new\_dir

FROM dueño

WHERE dni = new\_dni

INSERT INTO chofer (nombre, tel, dir, f\_licencia\_desde, f\_licencia\_hasta, dni)

VALUES (new\_nombre, new\_tel, new\_dir, f\_l\_d, f\_l\_h, new\_dni);

COMMIT;

END;

// DELIMITER ;

Id\_chofer no se pone al ser autoincremental