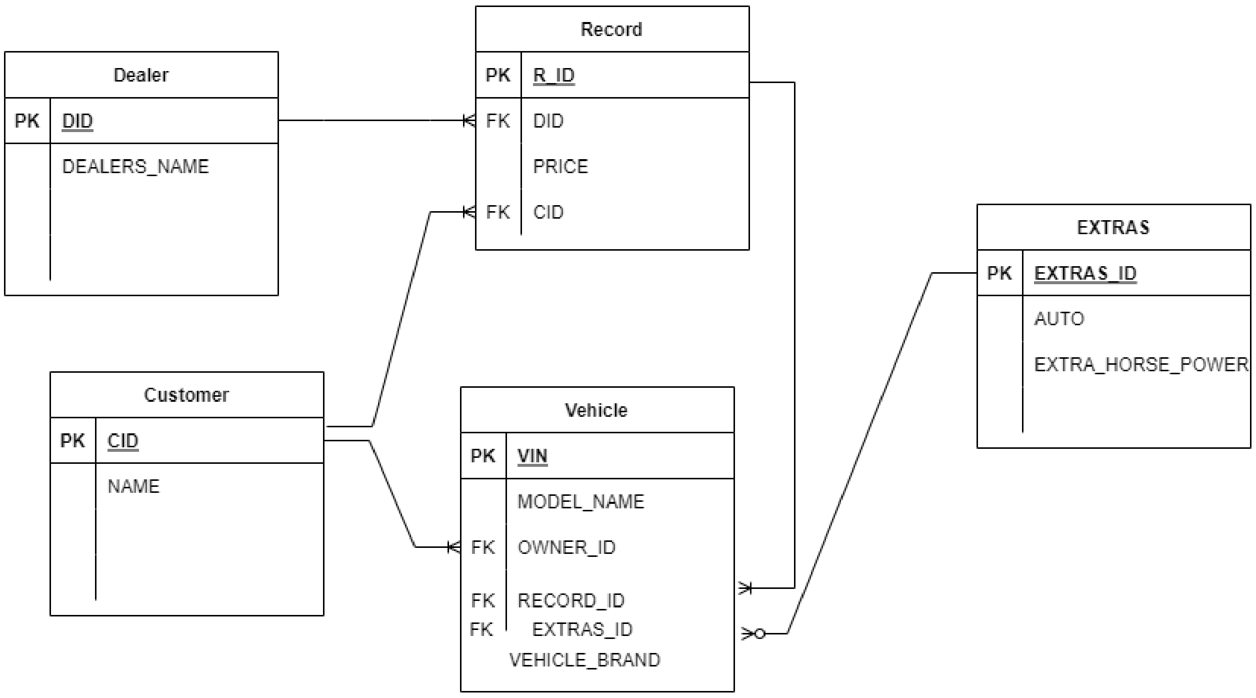
QUESTION 1 ER-DIAGRAM

At this first part of the project draw the desired ER-Diagram for an Automobile Company. The diagram present it below and also attached into deliverable with file name ER AUTOMOBILE COMPANY.png. As we can see the diagram apart from 5 entities and we have to mention also that the primary keys (PK) of each entity presented also at the table. Foreign keys (FK) also appeared on the diagram. The relation which seems on the diagram are 1toMany and 0toMany optional. I decided on Dealers table to store the id and the dealers name, on customer the corresponding features as dealer, in Record the records id, the dealers id who manage the sale, the price of the corresponding vehicle for customer and customers id. For the vehicle entity I create the features, Vin the owners id which specify by customers id who has the corresponding car, the record id of the car and extras which the corresponding vehicle has. Finally I keep the brands and model name for each vehicle. To conclude with Extras I create extras id and the features auto and extra_horse_power in order to specify that all cars could have for example auto(1) or extra horse power(1) or one of two of them or none meaning (0) from auto and (0) from horse power.



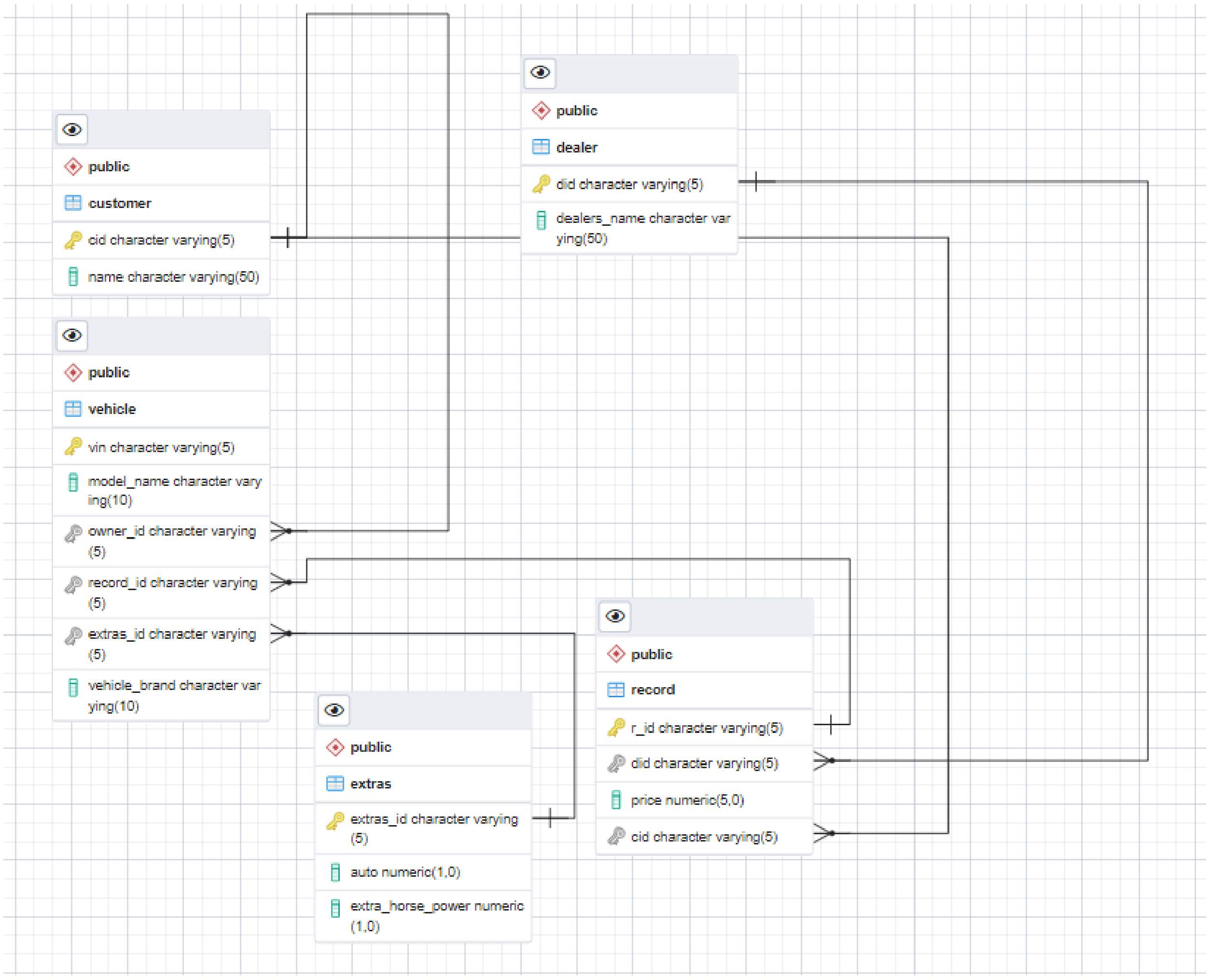
QUESTION 2

At the 2nd part we create the database and we insert in to corresponding tables some sample data. The implementation is reported at ask2db_inserts.sql attachment file.

The strategy was to simplify as much as possible each entities not only the number of tables but the number of inputs per table, without loss though the general goal which is was to create a database for an Automobile company.

I find much important for this task to add as Entities, the Vehicle, the Dealer, the Customer, Extras to specify if a Vehicle could offer with some extras but not all. And finally a Record table in order to keep in them some useful information about Vehicle Dealers and Customers.

I pay a lot attention in order to apply the properly constraints between the entities and of course observing the table creation, we also manage the delete and updates scenarios with on delete on update commands. The ERD which constructed by ADMIN4 is presented bellow.



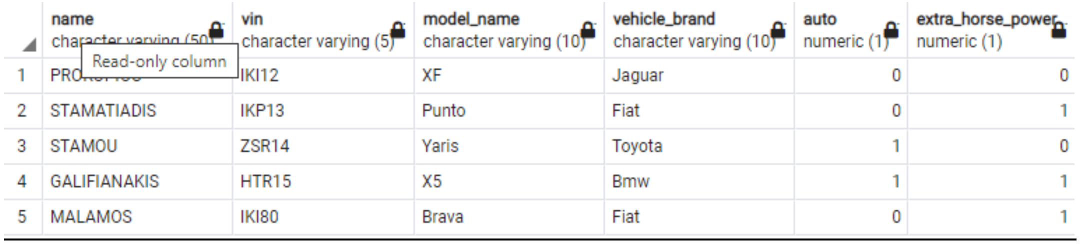
public
customer
cid character varying(5)
name character varying(50)
public
dealer
did character varying(5)
dealers_name character varying(50)
public
vehicle
vin character varying(5)
model_name character varying(10)
owner_id character varying(5)
record_id character varying(5)
extras_id character varying(5)
vehicle_brand character varying(10)
public
extras
extras_id character varying(5)
auto numeric(1,0)
extra_horse_power numeric(1,0)
public
record
r_id character varying(5)
did character varying(5)
price numeric(5,0)
cid character varying(5)

QUESTION 3

At the 3rd part we propose 2 queries statements that we believe our database going to requested many times to present the corresponding queries. Those 2 queries are reported on ask2Quey.sql attachment files. A small summary for each of them is:

QUERY1: We want to present the number of CUSTOMER(COUNT(*)) that a dealer has, the total sales (SUM(PRICE)) for each dealer and (DEALERS_NAME)

Output

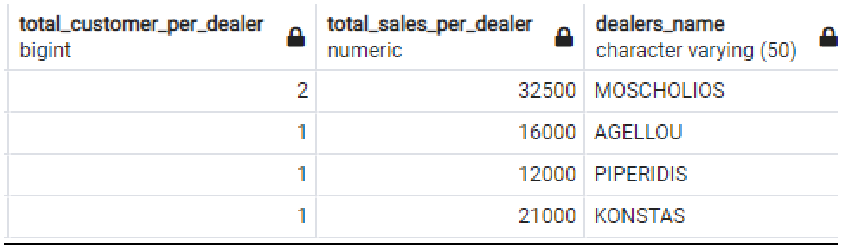


	name character varying (50)	vin character varying (5)	model_name character varying (10)	vehicle_brand character varying (10)	auto numeric (1)	extra_horse_power numeric (1)
1	PRO[illegible]	IKI12	XF	Jaguar	0	0
2	STAMATIADIS	IKP13	Punto	Fiat	0	1
3	STAMOU	ZSR14	Yaris	Toyota	1	0
4	GALIFIANAKIS	HTR15	X5	Bmw	1	1
5	MALAMOS	IKI80	Brava	Fiat	0	1

Read-only column

QUERY2: We want to present the CUSTOMERS NAME, his VIN and MODEL NAME the CORRESPONDING BRAND AND their EXTRAS

Output:



total_customer_per_dealer bigint	total_sales_per_dealer numeric	dealers_name character varying (50)
2	32500	MOSCHOLIOS
1	16000	AGELLOU
1	12000	PIPERIDIS
1	21000	KONSTAS

In order to construct the above queries I apply some inner joins between the corresponding tables and I create Views for each query to optimize the execute time. Furthermore, at the first query I construct a Materialized view in order to apply indexing and test how the execute time become. As the data are few the different using indexing or not is not significant.

QUESTION 4

At the last part the desired code was stored to attachment file PythonCode.ipynb. At the start we create some functions as menus for each desired option. In the sequel we open a connection with our database and start construct all of these cases. Depending, each time, the choice that user want to select the corresponding navigator menu appear and guide the user to communicate with our database. Also in update, in insertion and in deletion options we apply commit at the end in order to force the database to keep the changes. At the search option I decide to present the entities with most often requested which are Record and Vehicle. Both were constructed in order to searching using the corresponding ids for each table. Finally we close the connection to not allow anything exposed and leaked.