Logo, company name

Description automatically generated

**School of Computer Science**

**Masters in Applied Computing (M.A.C)**

**COMP8347**

**Internet Applications and Distributed Systems**

**Professor Dr Adel Abusitta**

**Lab Assignment**

**by**

**Hamza Baig (110089314)**

**Harsimran Singh (110090200)**

**Vivek Verma ()**

**Talha Muhammad Shamoon Chaudhry ()**

**Vidya Sehgal ()**

**Komalben Kishorbhai Patel (110091595)**

**Abhishek Singh (110089745)**

Table of Contents

[I - The new model **OrderVehicle** would look like: 4](#_Toc136865296)

[1. Vehicle serving as a foreign key: 4](#_Toc136865297)

[2. Buyer serving as a foreign key: 4](#_Toc136865298)

[3. Field that indicates number of vehicles being ordered: 4](#_Toc136865299)

[4. Field that indicates status: 4](#_Toc136865300)

[5. Field that indicates date order was updated: 4](#_Toc136865301)

[II – Add optional field in vehicles that describes product in few words: 4](#_Toc136865302)

[III – Add a phone number field in buyer: 4](#_Toc136865303)

[IV – Update default area of Buyer to Chatham: 4](#_Toc136865304)

[V- Remove fullname attribute from Buyer: 4](#_Toc136865305)

[VI – Add \_\_str\_\_ method in all models: 5](#_Toc136865306)

[VII – Write a method in OrderVehicle that returns the total price for all vehicles: 6](#_Toc136865307)

[Answer 1b: 7](#_Toc136865308)

[I – Admin.py 7](#_Toc136865309)

[II and III – Admin: 7](#_Toc136865310)

[IV - 7](#_Toc136865311)

[Answer 2: 10](#_Toc136865312)

[a. List the buyers having last name ‘Smith’: 10](#_Toc136865313)

[b. List the buyers whose addresses start with ‘444’: 10](#_Toc136865314)

[c. List the buyers who live on a ‘street’ in Windsor area: 10](#_Toc136865315)

[d. List the buyers who live in Chatham and Toronto: 10](#_Toc136865316)

[e. List the buyers who do not live in Windsor: 10](#_Toc136865317)

[f. List the buyers who are interested in CarType ‘Toyota’: 10](#_Toc136865318)

[g. List the vehicles that cost less than $30000: 10](#_Toc136865319)

[h. List the vehicles which are not available at the moment: 10](#_Toc136865320)

[i. List all the cartypes in which a buyer with username lara is interested in: 10](#_Toc136865321)

[j. List all the vehicles with a car\_price > $25000 and inventory <10: 10](#_Toc136865322)

[k. Get the first name of the buyer of the order having primary key (or pk) equal to 2: 10](#_Toc136865323)

[l. Write a query that first stores all cartypes in a variable called ‘all\_cartypes’, then calculates the length of ‘all\_cartypes’, and displays the third element of ‘all\_cartypes’: 11](#_Toc136865324)

[m. Write a query with a ‘for loop’ and a ‘print statement’ in it. You can choose any model (CarType, Vehicle, Buyer, OrderVehicle ) you like: 11](#_Toc136865325)

[Answer 3: 11](#_Toc136865326)

[Model Description with title, text, and created\_at: 11](#_Toc136865327)

[Perform Migrations: 12](#_Toc136865328)

[Admin.py: 12](#_Toc136865329)

[Add data for description table: 12](#_Toc136865330)

[Python Console: 12](#_Toc136865331)

[1. Get the first description from the description model (this answer should return a query set): 13](#_Toc136865332)

[2. Get the title of the first description Get the first description (this answer should return the text): 13](#_Toc136865333)

[3. Query all the database objects: 13](#_Toc136865334)

[4. From the Python Console, create a new description with a title and a description: 13](#_Toc136865335)

[5. Filter the description title based on the starting letters (ex. “this”): 13](#_Toc136865336)

[6. Filter the description that contains any word (ex. “Django”): 13](#_Toc136865337)

[7. Filter the description that does not contain any word (ex. “Django”): 13](#_Toc136865338)

[8. Filter the description that contains any word (ex. “Django”) but the title not having the same word: 14](#_Toc136865339)

Answer 1a

## I - The new model **OrderVehicle** would look like:

A picture containing text, electronics, screenshot, display

Description automatically generated

### Vehicle serving as a foreign key:

vehicle = models.ForeignKey(Vehicle, on\_delete=models.CASCADE)

### Buyer serving as a foreign key:

buyer=models.ForeignKey(Buyer, on\_delete=models.CASCADE)

### Field that indicates number of vehicles being ordered:

quantity = models.IntegerField()

### Field that indicates status:

status = models.IntegerField(choices=[(0, 'cancelled'), (1, 'placed'), (2, 'shipped'), (3, 'delivered')])

### Field that indicates date order was updated:

updated\_at = models.DateField(auto\_now=True)

## II – Add optional field in vehicles that describes product in few words:

description = models.TextField(blank=True)

## III – Add a phone number field in buyer:

phone\_number = models.CharField(max\_length=20, null=True,blank=True)

## IV – Update default area of Buyer to Chatham:

AREA\_CHOICES=[ ('W', 'Windsor'), ('LS', 'LaSalle'), ('A', 'Amherstburg'), ('L', 'Lakeshore'), ('LE', 'Leamington'), ('C','Chatham'), ('T','Toronto'), ('WL','Waterloo') ]

area = models.CharField(max\_length=2, choices=AREA\_CHOICES, default='C')

## V- Remove fullname attribute from Buyer:

Updated Buyer

A picture containing text, screenshot, display, font

Description automatically generated

## VI – Add \_\_str\_\_ method in all models:

A screen shot of a computer code

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, display, software

Description automatically generated

A picture containing text, screenshot, display, software

Description automatically generated

## VII – Write a method in OrderVehicle that returns the total price for all vehicles:

def total\_price(self):  
 return self.vehicle.car\_price \* self.quantity

# Answer 1b:

## I – Admin.py

A screen shot of a computer program

Description automatically generated with low confidence

## II and III – Admin:

A screenshot of a computer

Description automatically generated

## IV -

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

# Answer 2:

## List the buyers having last name ‘Smith’:

A picture containing text, font, screenshot

Description automatically generated

## List the buyers whose addresses start with ‘444’:

A picture containing text, font, screenshot, line

Description automatically generated

## List the buyers who live on a ‘street’ in Windsor area:



## List the buyers who live in Chatham and Toronto:



## List the buyers who do not live in Windsor:



## List the buyers who are interested in CarType ‘Toyota’:

A picture containing text, font, screenshot, line

Description automatically generated

## List the vehicles that cost less than $30000:



## List the vehicles which are not available at the moment:

A picture containing text, font, screenshot, line

Description automatically generated

## List all the cartypes in which a buyer with username lara is interested in:



## List all the vehicles with a car\_price > $25000 and inventory <10:



## Get the first name of the buyer of the order having primary key (or pk) equal to 2:

A screen shot of a computer

Description automatically generated with low confidence

## Write a query that first stores all cartypes in a variable called ‘all\_cartypes’, then calculates the length of ‘all\_cartypes’, and displays the third element of ‘all\_cartypes’:

A screen shot of a computer

Description automatically generated with low confidence

## Write a query with a ‘for loop’ and a ‘print statement’ in it. You can choose any model (CarType, Vehicle, Buyer, OrderVehicle ) you like:

A picture containing text, screenshot, font

Description automatically generated

# Answer 3:

## Model Description with title, text, and created\_at:

A picture containing text, screenshot, font

Description automatically generated

## Perform Migrations:

A picture containing text, screenshot, font

Description automatically generated

A screen shot of a computer program

Description automatically generated with low confidence

## Admin.py:

A screen shot of a computer

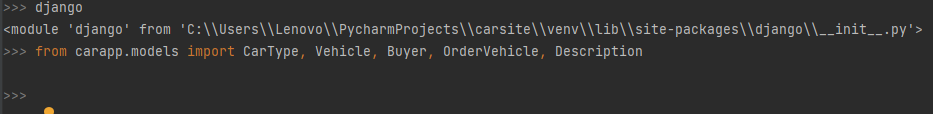
Description automatically generated with medium confidence

## Add data for description table:

A screenshot of a computer

Description automatically generated with medium confidence

## Python Console:



### Get the first description from the description model (this answer should return a query set):

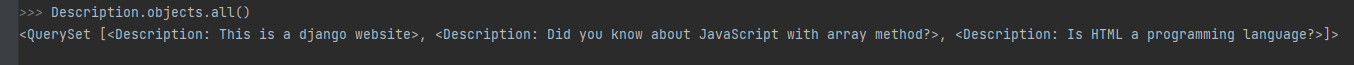


### Get the title of the first description Get the first description (this answer should return the text):





### Query all the database objects:



### From the Python Console, create a new description with a title and a description:



### Filter the description title based on the starting letters (ex. “this”):

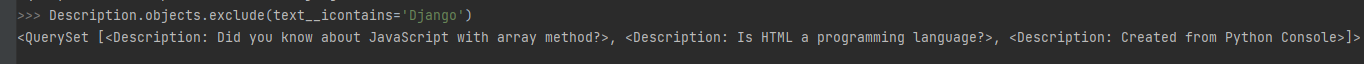


### Filter the description that contains any word (ex. “Django”):

icontains will not be case sensitive.



### Filter the description that does not contain any word (ex. “Django”):



### Filter the description that contains any word (ex. “Django”) but the title not having the same word:

 A screenshot of a computer

Description automatically generated with medium confidence