Academic Honesty

I fully support Baruch College's policy on Academic Honesty, which states, in part:

"Academic dishonesty is unacceptable and will not be tolerated. Cheating, forgery, plagiarism, and collusion in dishonest acts undermine the college's educational mission and the students' personal and intellectual growth. Baruch students are expected to bear individual responsibility for their work, to learn the rules and definitions that underlie the practice of academic integrity, and to uphold its ideals. Ignorance of the rules is not an acceptable excuse for disobeying them. Any student who attempts to compromise or devalue the academic process will be sanctioned."

Academic sanctions in this class will range from an F on the assignment to an F in this course. A report of suspected academic dishonesty will be sent to the Office of the Dean of Students. Additional information and definitions can be found at https://provost.baruch.cuny.edu/academic-affairs/academic_honesty/

To recognize your agreement with the Academic Honesty Policy, please **type** the following statement, type your CUNYID and Full Name, and date.

"The work in this assignment is my own. Any outside sources have been properly cited."

Statement: The work in this assignment is my own. Any outside sources have been properly cited

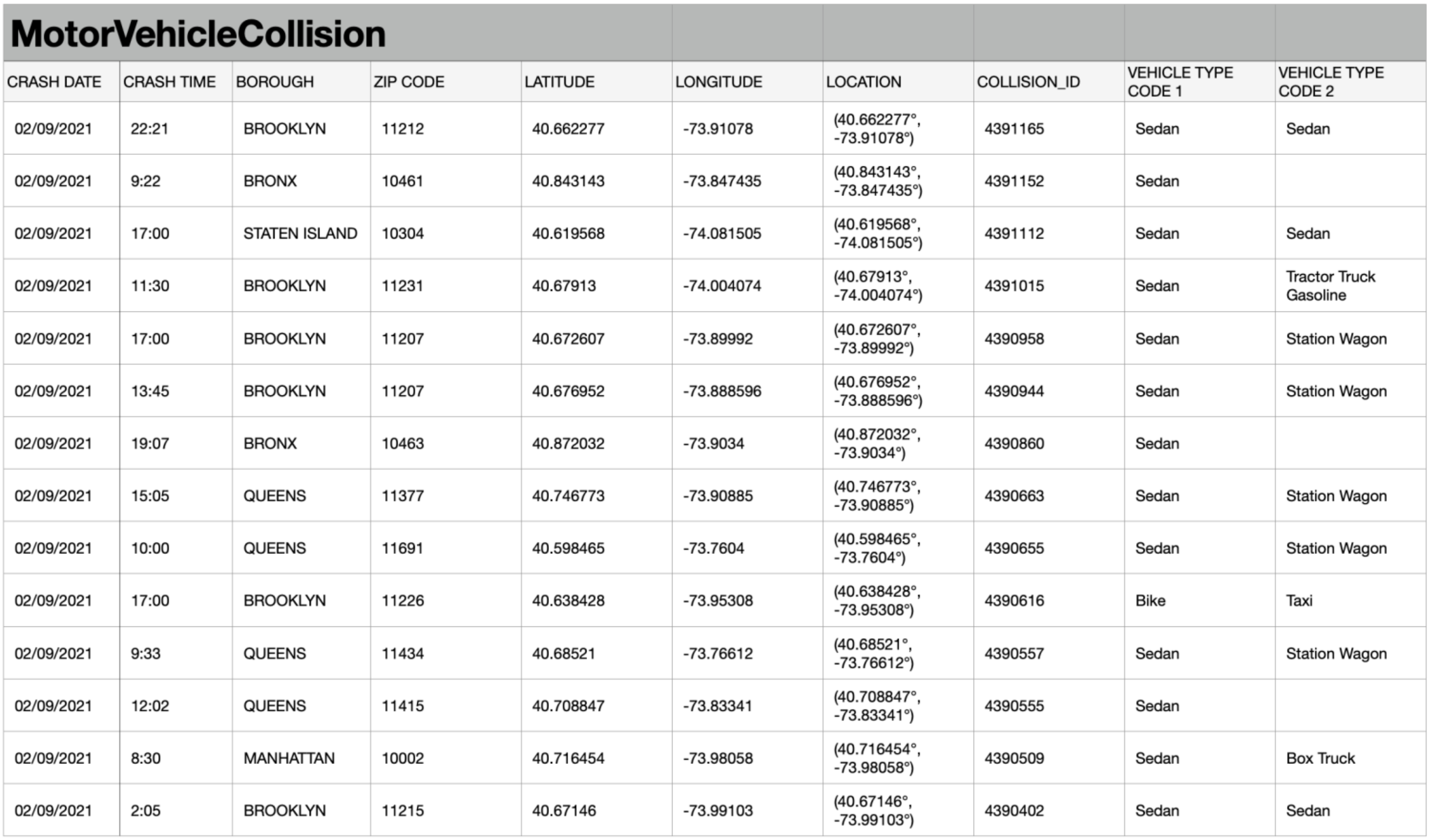
CUNYID: 23484732 First Name: William Last Name: Zhu Date: 6/28/2021

Instruction

- This assignment has 15 points in total which is 15% of the final grade.
- Please try your best to answer ALL questions.
- You must use Relational Schemas to represent the Relational Model. No credit will be granted for other notations.
  - Relational Schema: NAME(Attribute1, Attribute2, …, AttributeX(fk), … , AttributeN)
  - You should underline the primary key(s) and add (fk) for foreign key(s).
  - For each relation, list its functional dependencies separately.
- Print your completed work to a PDF file with name as: FirstnameLastnameA03.pdf
- Submit your work on time. No late submission will be accepted.

(3 points) Question 1: Functional Dependencies.

We have the following relation represented as a table. Read it carefully and answer following questions:



MotorVehicleCollision

CRASH DATE	CRASH TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	COLLISION_ID	VEHICLE TYPE CODE 1	VEHICLE TYPE CODE 2
02/09/2021	22:21	BROOKLYN	11212	40.662277	-73.91078	(40.662277°, -73.91078°)	4391165	Sedan	Sedan
02/09/2021	9:22	BRONX	10461	40.843143	-73.847435	(40.843143°, -73.847435°)	4391152	Sedan	
02/09/2021	17:00	STATEN ISLAND	10304	40.619568	-74.081505	(40.619568°, -74.081505°)	4391112	Sedan	Sedan
02/09/2021	11:30	BROOKLYN	11231	40.67913	-74.004074	(40.67913°, -74.004074°)	4391015	Sedan	Tractor Truck Gasoline
02/09/2021	17:00	BROOKLYN	11207	40.672607	-73.89992	(40.672607°, -73.89992°)	4390958	Sedan	Station Wagon
02/09/2021	13:45	BROOKLYN	11207	40.676952	-73.888596	(40.676952°, -73.888596°)	4390944	Sedan	Station Wagon
02/09/2021	19:07	BRONX	10463	40.872032	-73.9034	(40.872032°, -73.9034°)	4390860	Sedan	
02/09/2021	15:05	QUEENS	11377	40.746773	-73.90885	(40.746773°, -73.90885°)	4390663	Sedan	Station Wagon
02/09/2021	10:00	QUEENS	11691	40.598465	-73.7604	(40.598465°, -73.7604°)	4390655	Sedan	Station Wagon
02/09/2021	17:00	BROOKLYN	11226	40.638428	-73.95308	(40.638428°, -73.95308°)	4390616	Bike	Taxi
02/09/2021	9:33	QUEENS	11434	40.68521	-73.76612	(40.68521°, -73.76612°)	4390557	Sedan	Station Wagon
02/09/2021	12:02	QUEENS	11415	40.708847	-73.83341	(40.708847°, -73.83341°)	4390555	Sedan	
02/09/2021	8:30	MANHATTAN	10002	40.716454	-73.98058	(40.716454°, -73.98058°)	4390509	Sedan	Box Truck
02/09/2021	2:05	BROOKLYN	11215	40.67146	-73.99103	(40.67146°, -73.99103°)	4390402	Sedan	Sedan

List the functional dependencies in the relation above.

4 total fds

Collision_ID -> CrashDate, CrashTime, Borough, ZipCode, Latitude, Longitude, Location, VehicleTypeCode1, VehicleTypeCode2

Latitude,Longitude -> Location

Borough -> Zipcode

CrashDate -> CrashTime

(4 points) Question 2: We have a relational model represented as a relational schema and its functional dependencies given as below:

CustomerContact (FirstName, LastName, DoB, Email, Zodiac, Street, County, City, State, Zip, Company, CompanyInfo)
fd1: FirstName, LastName, DoB -> Email, Zodiac, Street, County, City, State, Zip, Company, CompanyInfo
fd2: FirstName, DoB -> Email
fd3: DoB -> Zodiac
fd4: Zip -> City, State
fd5: Company -> CompanyInfo

1) Is this relational model in 1NF? Why? If not, normalize it to 1NF. Make sure you list ALL relations after the normalization process; and for each relation, list the functional dependencies.
Yes it is in 1NF

2) Is this relational model in 2NF? Why? If not, normalize it to 2NF. Make sure you list ALL relations after the normalization process; and for each relation, list the functional dependencies.

No because there is a partial dependency with
fd2: FirstName, DoB -> Email
fd3: DoB -> Zodiac

we need to create a new relation and remove the duplicates from CustomerContact
DoB_Zodiac (DoB, zodiac,)
FD1: DoB -> Zodiac

FirstName_DoB_Email(FirstName, DoB, Email

fd1: FirstName, DoB -> Email

So we have:

CustomerContact (FirstName(fk), LastName, DoB(fk), Email, Zodiac, Street, County, City, State, Zip, Company, CompanyInfo)
fd1: FirstName, LastName, DoB ->Street, County, City, State, Zip, Company, CompanyInfo
fd2: Zip -> City, State
fd3: Company -> CompanyInfo

FirstName_DoB_Email(FirstName, DoB, Email

fd1: FirstName, DoB -> Email

DoB_Zodiac (DoB, zodiac)
FD1: DoB -> Zodiac

3) Is the relational model in 3NF? Why? If not, normalize it to 3NF. Make sure you list ALL relations after the normalization process; and for each relation, list the functional dependencies.

No it is not in 3NF because there is transitive dependency with
Fd2: Zip -> City, State
Fd3: Company -> CompanyInfo

To normalize this relation into 3NF, we create new relations.
Zip_City_State(Zip, City, State)
FD1: Zip-> City, State

Company_CompanyInfo(Company, CompanyInfo)
Company-> Company Info

Finally we have:
CustomerContact (FirstName(fk), LastName, DoB(fk), Email, Zodiac, Street, County, Zip (fk), Company(fk))
fd1: FirstName, LastName, DoB ->Street, County, Zip , Company

Zip_City_State(Zip, City, State)
FD1: Zip-> City, State

Company_CompanyInfo(Company, CompanyInfo)
Company-> Company Info

FirstName_DoB_Email(FirstName, DoB, Email

fd1: FirstName, DoB -> Email

DoB_Zodiac (DoB, zodiac)
FD1: DoB -> Zodiac

(8 points) Question 3: We have a relational model represented as a relational schema and its functional dependencies given as below:

Assignments(A, B, C, D, E, F, G, H, I, J, K, L, M, O)
fd1: A, B, C -> D, E, F, G, H, I, J, K, L, M, O

fd2: A -> D, E, F, L, M, O
fd3: B, C -> G, H
fd4: D -> L, M, O
fd5: L -> M
fd6: G -> H

1) Is this relational model in 1NF? Why? If not, normalize it to 1NF. Make sure you list ALL relations after the normalization process; and for each relation, list the functional dependencies.
Yes

2) Is this relational model in 2NF? Why? If not, normalize it to 2NF. Make sure you list ALL relations after the normalization process; and for each relation, list the functional dependencies.
No because there is partial dependency with FD2, FD3 so we create new relations
A_DEFLMO(A, D, E, F, L, M, O)
FD1: A-> D, E, F, L, M, O

BC_GH(B,C, G, H)
FD1: B,C -> G, H

Finally,

Assignments(A(fk), B(fk), C(fk), I, J, K, L, M, O)
fd1: A, B, C -> D, E, F, G, H, I, J, K, L, M, O
fd2: D -> L, M, O
fd3: L -> M
fd4: G -> H

A_DEFLMO(A, D, E, F, L, M, O)
FD1: A-> D, E, F, L, M, O

BC_GH(B,C, G, H)
FD1: B,C -> G, H

3) Is the relational model in 3NF? Why? If not, normalize it to 3NF. Make sure you list ALL relations after the normalization process; and for each relation, list the functional dependencies.

No because there is transitive dependency with FD3(fd3: L -> M) and D-> L, M, O. So we create a new relation

L_M(L, M)
Fd1: L -> M

D_LMO(D, L, M, O)
FD1: D->L, M, O

Finally:

Assignments(A(fk), B(fk), C(fk), D(fk), I, J, K, L(fk), M, O)
fd1: A, B, C -> D, E, F, G, H, I, J, K, L, M, O
fd4: G -> H

A_DEFLMO(A, D, E, F, L, M, O)
FD1: A-> D, E, F, L, M, O

BC_GH(B,C, G, H)
FD1: B,C -> G, H

L_M(L, M)
Fd1: L -> M

D_LMO(D, L, M, O)
FD1: D->L, M, O