Assignment 3 NORMALIZATION
Hua Guo hxg142430

Total points: 50

This assignment should be completed individually. Save your completed assignment as a word document or .pdf file and submit via eLearning.

Complete the two Normalization problems below (A & B). The final product must be your own work. In your solution, show the Initial Grouping, 1NF, Dependency Diagram, 2NF, and 3NF. When appropriate, be sure to identify all functional dependencies (both partial and transitive).

Part A: TRIP

The table structure shown in below contains many unsatisfactory components and characteristics. For example, there are several multivalued attributes, naming conventions are violated, and some attributes are not atomic.

Sample TRIP Records

Attribute Name	*Sample Value*	*Sample Value*	*Sample Value*	*Sample Value*
TRIP	1003	1018	1019	1023
DATE	15-Jan-2008	15-Jan-2008	16-Jan-2008	17-Jan-2008
CITY	STL	MIA	TYS	ATL
MILES	580	1,290	524	768
CUST_NUM	10232	10233	10234	10235
CUST_LNAME	Brown	Hanson	Bryana	Brown
PASSANGERS	5	12	2	5
CARGO	235 lbs.	18,940 lbs.	348 lbs.	155 lbs.
PILOT	Smith	Chen	Henderson	Smith
COPILOT		Henderson	Smith	
FLT_ENGINEER		O'Shaski		
LOAD_MASTER		Davis		
AC_NUMBER	1234Q	3456Y	1234Q	2256W
PLANE_CODE	PA31-350	CV-580	PA31-350	PA31-350
PLANE_SEATS	10	38	10	10
PLANE_CHG_MILE	$2.79	$23.36	$2.79	$2.79

1. **Write the relational schema and list the dependencies for the table structure. Make sure that you label all dependencies. The CHAR_MILES entry is based on round-trip miles, including pickup points. (Hint: Look at the data values to determine the nature of the relationships. For example, note that employee Smith has flown two charter trips as pilot and one trip as copilot.) (10 points)**

   See example on the last page on how to write this.

   Assumptions :
   There is no more information about whether the trip is routine or unique. Here, I assume the tip is unique.

   Relational schema:

   (**TRIP**, DATE, CITY, MILES, CUST_NUM, CUST_LNAME, PASSANGERS, CARGO, PILOT, COPILOT, FLT_ENGINEER, LOAD_MASTER, AC_NUMBER, PLANE_CODE, PLANE_SEATS, PLANE_CHG_MILE)

   Primary Key:

   TRIP

   Dependencies:

   (TRIP → CUST_NUM, AC_NUMBER, DATE, CITY, MILES, CUST_LNAME, PASSANGERS, CARGO, PILOT, COPILOT, FLT_ENGINEER, LOAD_MASTER, PLANE_CODE PLANE_SEATS, PLANE_CHG_MILE)

   Partial Dependencies:
   No Partial Dependencies

   Transitive Dependencies
   CUST_NUM → CUST_LNAME
   AC_NUMBER → PLANE_CODE, PLANE_SEATS, PLANE_CHG_MILE
   PLANE_CODE → PLANE_SEATS, PLANE_CHG_MILE

2. **Convert the relation to 3NF. (*Hint*: You might have to create a few new attributes. Also make sure that the new dependency diagrams contain attributes that meet proper design criteria; that is, make sure there are no multivalued attributes and that the naming conventions are met.) Write the corresponding relational schemas. Identify each primary key and each foreign key. (10 points)**

   (Assumptions:

TRIP (**TRIP_CODE**, DATE, CITY, MILES, CUST_NUM, PASSANGERS, CARGO, AC_NUMBER , FLT_ENGINEER, LOAD_MASTER)

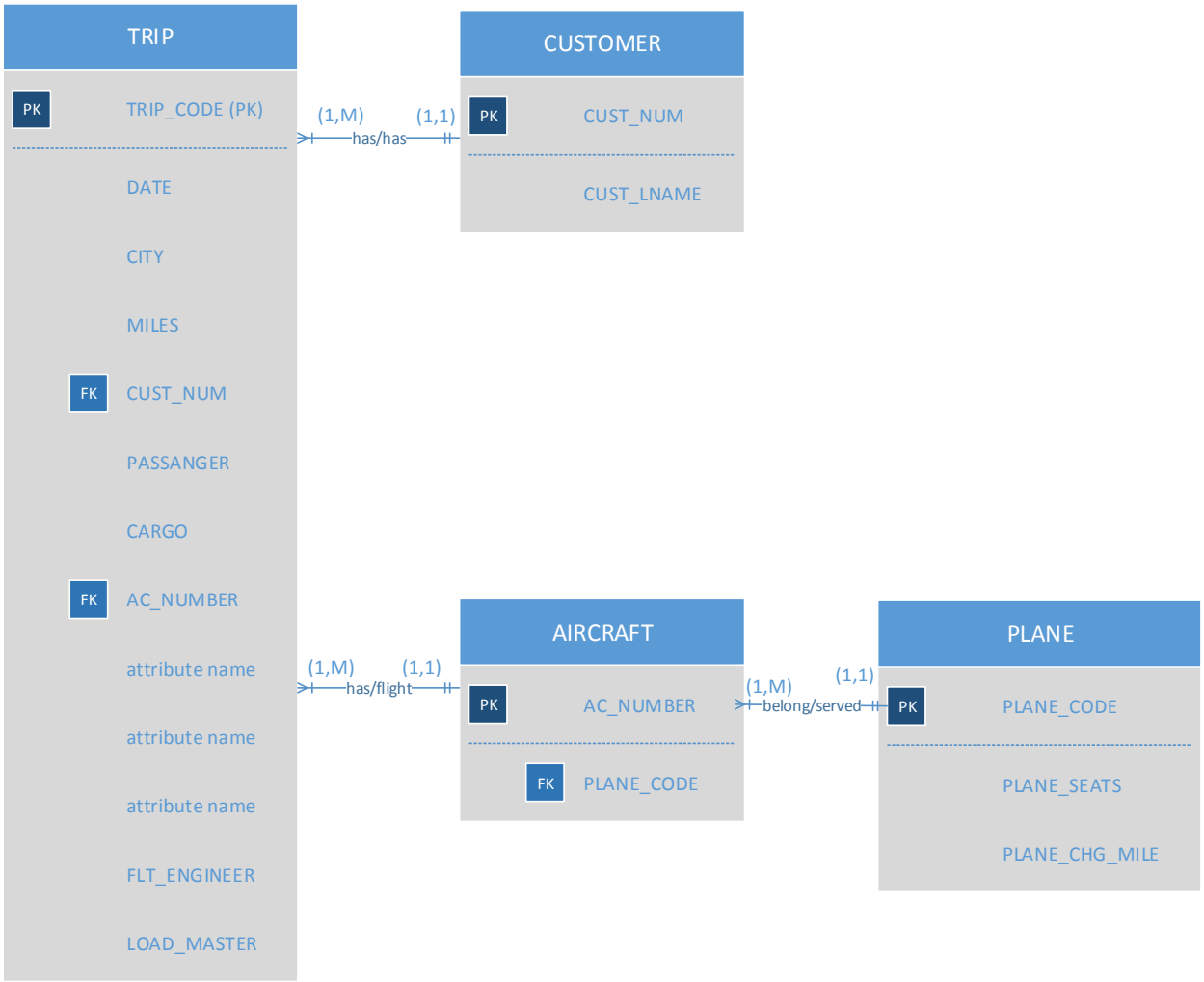
CUSTOMER (**CUS_NUM**, CUS_LNAME)

AIRCRAFT (**AC_NUMBER**, PLANE_CODE)

PLANE (**PLANE_CODE**, PLANE_SEATS, PLANE_CHG_MILE)

3. Draw the ER Diagram. (5 points)

Assumptions:
When one person buy a ticket, the person becomes a customer. So a customer has one or more trip.
One trip has one and only one aircraft. This means a given aircraft can served for one or more trip.
A given model of plane can be sold to serve as one or more aircraft.



Part B: WAREHOUSE

The warehouse database stores inventory information. A warehouse has one manager who has an employee number (unique) and a name. A warehouse is identified by its warehouse name (unique) and its address. Every part in the warehouse has a part number (unique), inventory date, quantityon-hand, supplier name (one), and the delivery number, data, and quantity. One delivery could deliver several parts. The database is shown below:

WAREHOUSE (manager-id, manager-name, warehouse-name, warehouse-address, part-no, inventory-date, qty-on-hand, supplier-name, delivery-no, delivery-date, delivery-qty)

1. **List the primary key(s) and list all transitive and/or partial dependencies in the relational schema diagram provided. (10 points)**

   See example on the last page on how to write this.

   In this question, I want to use MANAGER_ID, MANAGER_NAME, WAREHOUSE_NAME, WAREHOUSE_ADDRESS, PART_NO, INVENTORY_DATA, QTY_ON_HAND, SUPPLIER_NAME, DELIVERY_NO, DELIVERY_DATE, DELIVERY_QTY in part 2 and part 3, instead of the same content used in the above question description.

   Primary Key:

   delivery-no, warehouse-name, part-no

   Dependencies:

   (warehouse-name, part-no, delivery-no → manager-id, manager-name, warehouse-address, inventory-date, qty-on-hand, supplier-name, delivery-no, delivery-date, delivery-qty)

   Partial Dependencies:
   warehouse-name, part-no → inventory-date, qty-on-hand, supplier-name

   part-no → inventory-date, qty-on-hand, supplier-name

   delivery-no → delivery-date, delivery-qty

   warehouse-name → warehouse-address

   Transitive Dependencies

   manager-id→ manager-name

2. **Convert the relation to 3NF. (*Hint*: You might have to create a few new attributes. Also make sure that the new dependency diagrams contain attributes that meet proper design criteria; that is, make sure there are no multivalued attributes and that the naming conventions are met.) Write the corresponding relational schemas. Identify each primary key and each foreign key. (10 points)**

(Assumptions:
In this question, we have no more information about how many kinds of part one supplier can supplied. One supplier would supply one or several parts. Here, I assume one supplier only supplies one part.)

EMPLOYEE (**MANAGER_ID**, MANAGER_NAME)

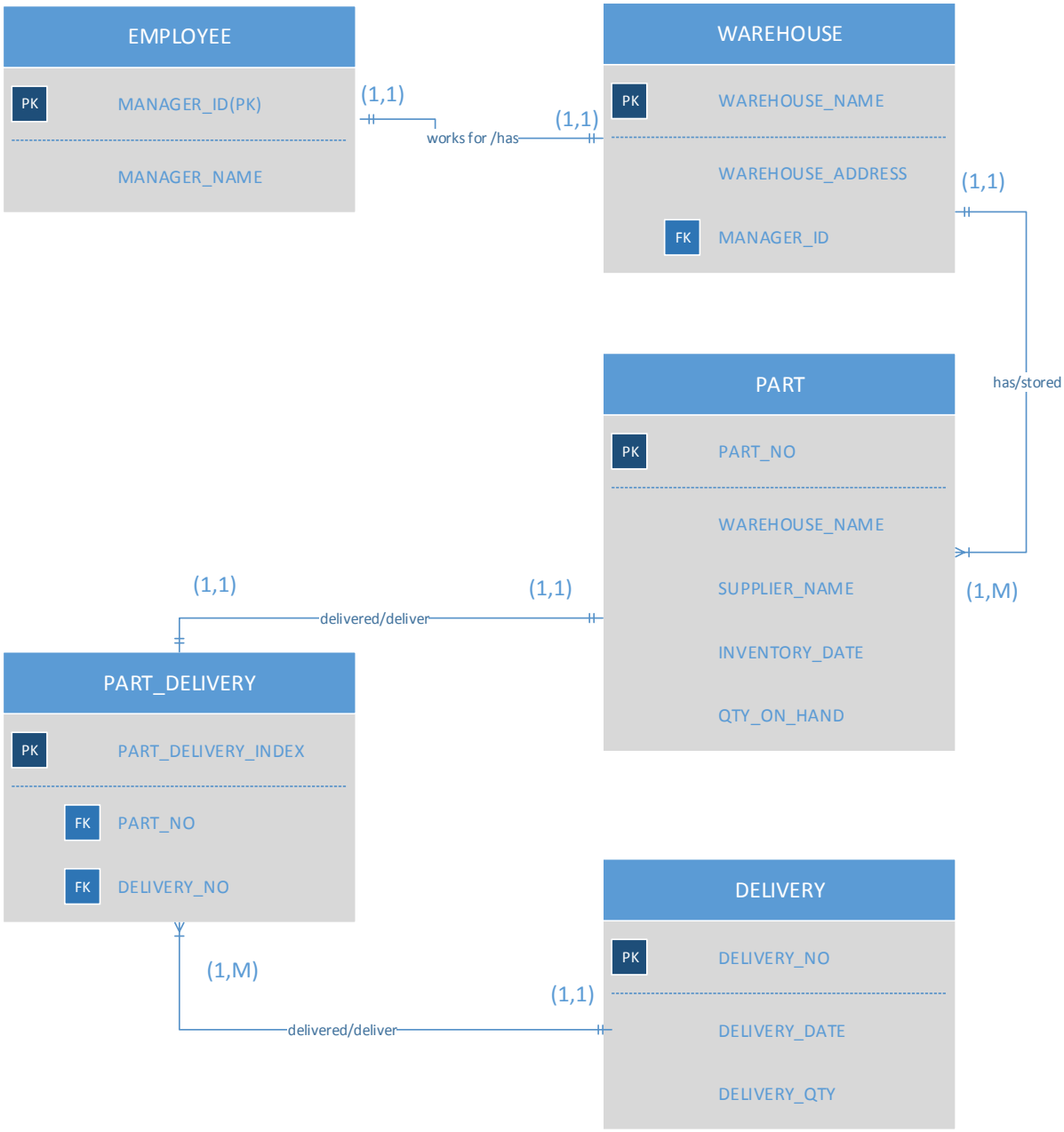
WAREHOUSE (**WAREHOUSE_NAME**, WAREHOUSE_ADRESS, MANAGER_ID)

PART (**PART_NO**, DELIVERY_NO, WAREHOUSE_NAME, SUPPLIER_NAME, INVENTORY_DATE, QTY_ON_HAND)

DELIVERY (**DELIVERY_NO**, DELIVERY_DATE, DELIVERY_QTY)

PART_DELIVERY (**PART_DELIVERY_INDEX**, PART_NO, DELIVERY_NO)

3. **Draw the ER Diagram. (5 points)**
   Assumptions:
   Because the part number is unique. So, I suppose *one part only stored in one warehouse.*


EMPLOYEE
PK
MANAGER_ID(PK)
MANAGER_NAME
(1,1)
works for /has
(1,1)
WAREHOUSE
PK
WAREHOUSE_NAME
WAREHOUSE_ADDRESS
FK
MANAGER_ID
(1,1)
has/stored
PART
PK
PART_NO
WAREHOUSE_NAME
SUPPLIER_NAME
INVENTORY_DATE
QTY_ON_HAND
(1,M)
(1,1)
delivered/deliver
(1,1)
PART_DELIVERY
PK
PART_DELIVERY_INDEX
FK
PART_NO
FK
DELIVERY_NO
(1,M)
(1,1)
delivered/deliver
DELIVERY
PK
DELIVERY_NO
DELIVERY_DATE
DELIVERY_QTY

Example:

Relational Schema
EMPLOYEE(**EMP_CODE**, EMP_LNAME, EMP_EDUCATION, JOB_CLASS,
EMP_BASE_SALARY, EMP_COMMISSION_RATE)

The **dependencies** shown as:

EMP_CODE → emp_lname, emp_education, job_class….

Partial Dependencies
No Partial Dependencies

Transitive Dependencies
JOB_CLASS → emp_base_salary