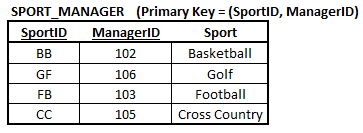
William Eddy

Data Management - Fall 2018

HW C

**Part A (Normalization)**

Look at each example below and indicate which normal form is violated and why it is violated and then rewrite the attribute list to correct it. Primary keys in the examples are underlined. Be sure to underline any primary key(s) and italicize any foreign keys.

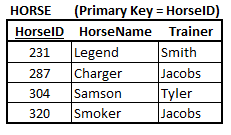
1. 

**Normal Form Violated in the table above: \_\_\_\_\_\_\_\_\_\_\_\_\_2nd\_\_\_\_\_**

SOLUTION:

SPORT (*SportID*, Sport)

MANAGER (*SportID,* ManagerID)

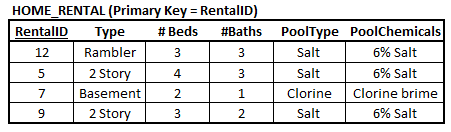


**Normal Form Violated in the table above: \_\_\_\_\_\_3rd\_\_\_\_\_\_\_\_**

SOLUTION:

HORSE (*HorseID*, HorseName)

TRAINER (*HorseID*, Trainer)

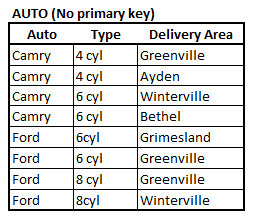
1. 

**Normal Form Violated in the table above: \_\_\_\_\_\_3rd\_\_\_\_\_\_\_\_\_\_\_\_**

SOLUTION:

RENTALS (*RentalID,* Type, #Beds, #Baths)

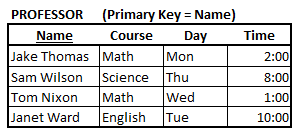
POOLS (*RentalID*, PoolType, PoolChemicals)

1. 

**Normal Form Violated in the table above: \_\_\_\_1st\_\_\_\_\_\_\_\_\_\_\_**

SOLUTION:

CARS (CarID, Auto, Type, DeliveryArea)

1. 

**Normal Form Violated in the table above: \_\_\_\_\_\_\_\_\_1st\_\_\_\_\_\_\_\_\_\_\_\_**

SOLUTION:

COURSES (ClassNumber, Course, Name, Day, Time)

**Part B**

Consider the following fields: (listed in alpha order)

LeagueID, LeagueName, ManagerLName, ManagerFName, PlayerID, PlayerLName, PlayerFName, Schedule, SportID, SportName, Stats, TeamID, TeamName

Consider the following statements:

* Players are part of a team
* A team plays a certain sport and is part of a league

1. Put the above named fields into 4th Normal Form using the following steps:
   1. Show all functional dependencies

PlayerLName -> PlayerFName -> PlayerID -> Team -> ManagerLName -> ManagerFName ->

TeamName -> SportName -> LeagueName -> LeagueID

TeamName -> Schedule

TeamID -> Team -> Stats

* 1. Identify every candidate key

(PlayerFName, PlayerLName)

(PlayerID)

* 1. To make things easier, convert each of the above into a Sentence Structure

LEAGUES (LeagueID, LeagueName, *SportID*)

SPORTS (*SportID*, SportName)

TEAMS (*TeamID*, *SportID*, TeamName, Schedule)

MANAGERS (ManagerID, ManagerFName, ManagerLName, *TeamID*)

PLAYERS (PlayerID, PlayerFName, PlayerLName, Stats, *TeamID*)

* 1. Put each into 4NF breaking out any fields, if needed. Be sure to underline the primary key field of each relation and indicate the *foreign key* if any.

LEAGUES (LeagueID, LeagueName, *SportID*)

SPORTS (*SportID*, SportName)

TEAMS (*TeamID*, *SportID*, TeamName, Schedule)

MANAGERS (ManagerID, ManagerFName, ManagerLName, *TeamID*)

PLAYERS (PlayerID, PlayerFName, PlayerLName, Stats, *TeamID*)

**Part C**

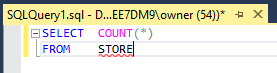
**Please note**: On all SQL requests the tables are NOT in the server BUT you should still go into SQL (using the seesaw2 server) and write the code and then paste it into the document where requested.

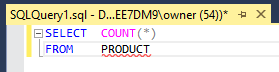
You are in receipt of the following two new tables of data:

**STORE** (StoreNumber, StoreLocation, StorePhone)

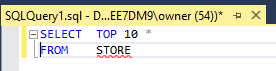
**PRODUCT** (ProdNumber, ProdName, ProdPrice, ProdCost, *Store#*)

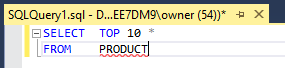
1. You are asked to create a new database using the two new tables above. To do this you must first access the table structure above to determine if it is correct and valid as follows:
   1. Count the number of rows in each table above using the COUNT function showing the code below.



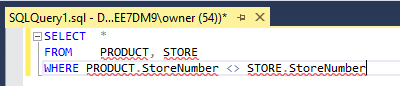


* 1. Select the first 10 records of each table showing the code below.





* 1. Since it is risky to assume that referential integrity constraints have been enforced between the two tables write the SQL code to determine if the constraint has been enforced.



* 1. What result should you get if the referential integrity statement has been enforced?

No results

1. You are now ready to create the new database. You have decided to combine them into just one table named STORE\_PRODUCT because it will be for decision making only and not a transaction based database. Write the code below to set the new table up assuming there are already two tables named STORE and PRODUCT in the database. Assign a surrogate key as a primary key instead of having a composite primary key.

ItemID is assigned as the surrogate key (you can use any named field).

