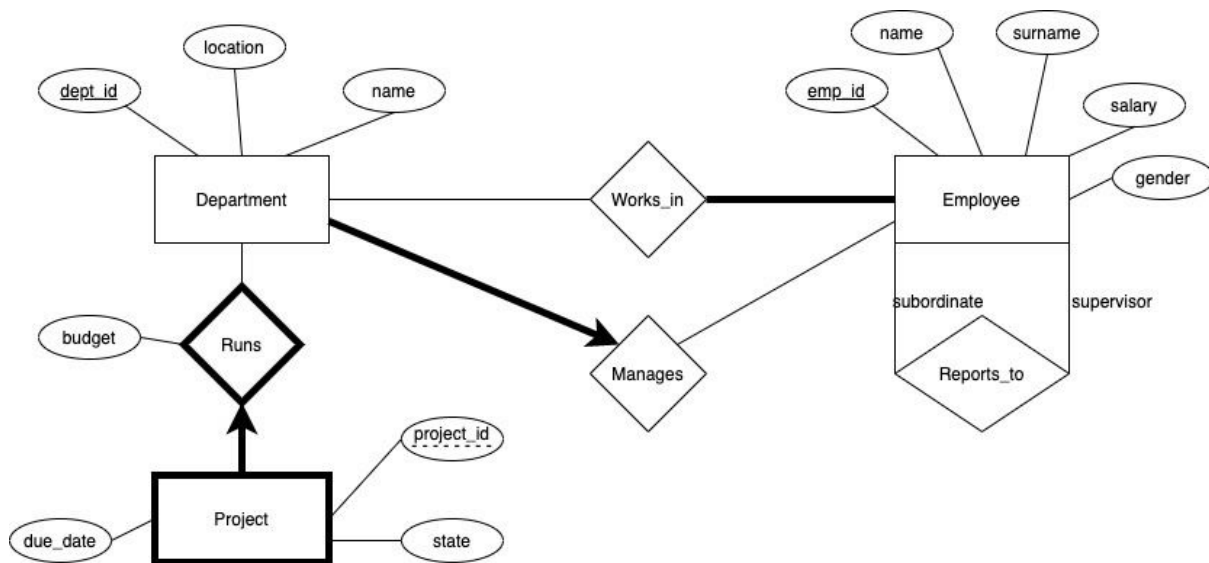# Ceng352 - Database Management Systems
Written Assignment 1

2019/2

1. Consider the following E/R diagram:

a) Convert the E/R diagram to relational tables (provide SQL statements). Don't forget to specify the primary key and foreign keys. For foreign key constraints, assume the following:
- If an employee is deleted all the records on work in a department relation must be also deleted.
- If an employee is deleted the corresponding managers in manages relationship should be set to the default manager "101"
- An attempt to delete a department must be prohibited if an employee works in that department.

b) Note that there is a total participation constraint in the E/R diagram stating that every employee must be working in at least one department (i.e. the line between Works_in and Employee is thick and bold). Write an assertion called Total to enforce this constraint.

c) Define appropriate SQL CHECK clauses to implement the following constraints:
   ● Employee salary must be greater than the minimum wage, which is 36000.
   ● Department names must include the location of the department (For example R&D department which is located in "IST" has the name "ISTR&D" or "Sales" department in "ANK" has the name "SalesANK").

d) Define an AFTER trigger to implement the following constraint on the Project table: When the budget of a project is updated, if it is decreased then set Project state to "Unsuccessful".

2. For the given E/R diagrams, find the maximum number of tuples that can be stored in the relation R.

a) Assume that the Product table consists of 100 rows with unique id's, the Store table has 5 rows with unique id's and the Person table contains 1000 rows with unique id's.
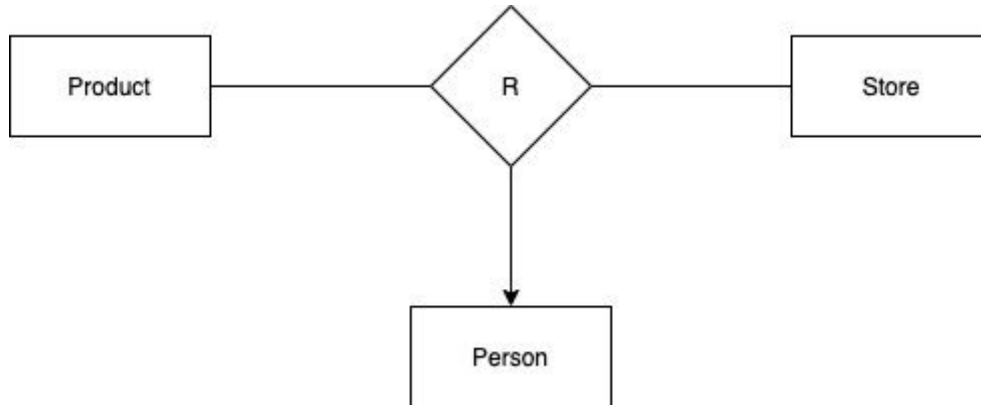
b) Assume that the Product table consists of 100 rows with unique id's, the Store table has 5 rows with unique id's and among 1000 Person entries there are 10 SalesPerson.
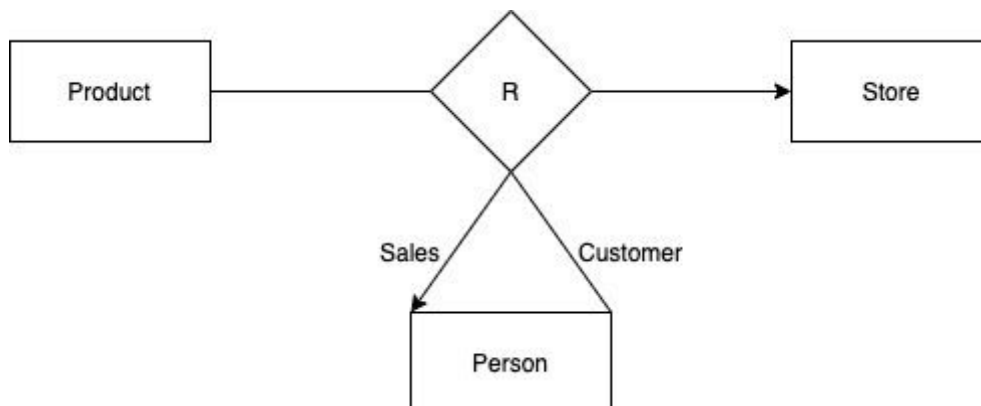
3. Given the schema R = {A, B, C, D, E, F, G, H} and the functional dependencies below:

$$\{A \rightarrow C, B \rightarrow E, CB \rightarrow F, FE \rightarrow G, FG \rightarrow AH\}$$

Prove the following inference rules using Armstrong's Axioms. Show all your work (instincts, assumptions are not valid). For each of the rule you have used, give the name of it.

a) CB → G
b) AB → EF

4. Given the schema R = {A, B, C, D, E, F, G} and the functional dependencies below:

$$\{A \rightarrow B, CD \rightarrow E, F \rightarrow D, E \rightarrow G, AC \rightarrow D, D \rightarrow C\}$$

   a) Find all keys (Show your work).
   b) Is R in BCNF or not? Explain why.
   c) If it is not in BCNF form decompose it into a collection of BCNF relations.
   d) Show that the above decomposition is

   i) dependency-preserving or not?

   ii) lossless-join or not?

5. You are given a ".csv" file which contains a table for a simple database. The table has some data anomalies due to redundancy. Your task is to load this table into PostgreSQL and identify the functional dependencies that cause anomalies by writing SQL statements. Once you identify the "bad" functional dependencies, your task is to normalize the table.

Do the following:

   (1) Download and install PostgreSQL if you haven't done so far.
   (2) Create a table in the database and load the table with the data given in the given ".csv" file. You can use the "COPY" property of PostgreSQL. You can get detailed information from here.
   (3) Find all functional dependencies in the table by writing appropriate SQL queries. Remember that a functional dependency is a constraint on a database instance. First, try to identify simple FDs like A → B, then try AB → C, etc. You should write an SQL query for each candidate FD. You can see if the FD holds or not by checking the answer of the query.
   (4) Decompose the table into BCNF tables using the FDs that you discovered. Create tables for normalized relations. Don't forget to create keys and foreign keys for the BCNF schema.
   (5) Load the new tables with the data from the original table. For this step, you should write SQL statements to load data into the new tables.
   (6) Dump created database into 'eXXXXXXX.sql' file. You can dump the database using 'pg_dump'. Detailed information can be found here.

What to turn in:

   a) List of all FDs you identified and the corresponding SQL queries to discover them at the end of step 3 above.
   b) List of all SQL statements to create normalized tables.
   c) List of all SQL statements that load the contents of the tables.
   d) Send 'eXXXXXXX.sql' file along with the pdf.

Submission

You should send a pdf file, named 'eXXXXXXX.pdf' (your seven-digit ID number) contains your answers. You can prepare the pdf using both 'DOCS' or 'Latex', doesn't matter. Moreover, you can send a scanned version of your handwritten answers, but please be sure that it is readable.

For Question 5, if you completed all of the steps you should have created an 'eXXXXXXX.sql' file (your seven-digit ID). Compress both '.pdf' file and '.sql' file to 'eXXXXXXX.zip' file ('.rar' or '.tar' are also accepted) and upload it.