

CSE 5330/7330 Fall 2017

Project Definition

Do your own work. Reference any material used.

(Alternative Projects will be considered by Dr. Moore. Please contact him with suggestions.)

This is an individual project to be completed by each student. In addition, the project is to be completed in 3 phases. At the end of each phase, you are to submit that portion of the project for grading. At that time, you will receive information needed to complete the next phase of the project.

*You are to design and implement a database system to keep track of software builds performed by **Our Software Factory (OSF)**. Your user contact (and person to approach with any questions) is Dr. Moore.*

Any DBMS may be used for the project as long as a SQL command line interface is available for customer acceptance testing. The last phase of the project will require that Dr. Moore be able to test some unseen queries against your database, thus he must have access to your database. If you use the SMU implementation of Oracle or MySQL, this can easily be accomplished using appropriate security settings. If, however, you use another DBMS then you will have to provide instructions for its use or schedule a time for hands-on interaction.

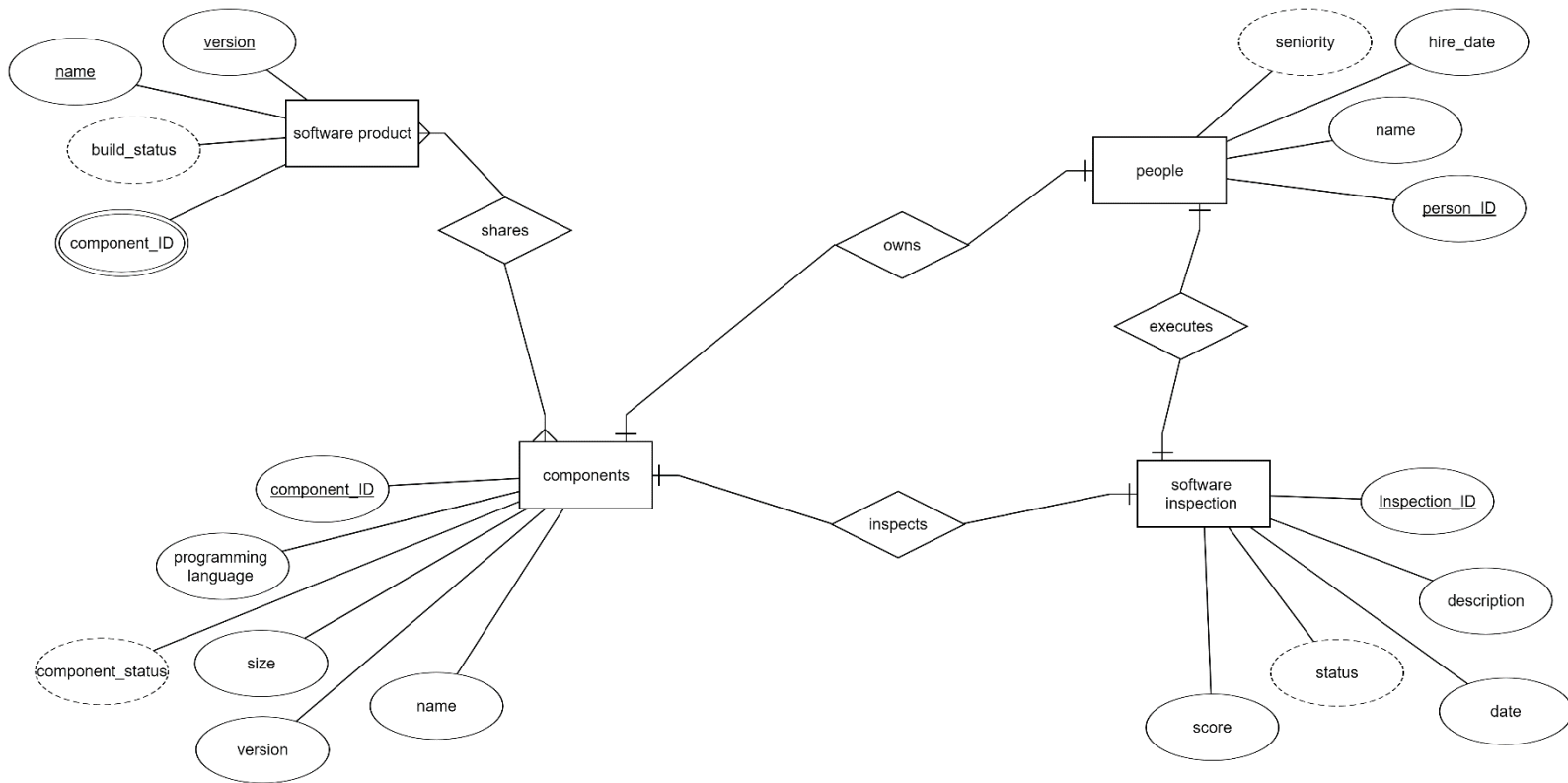
Our Software Factory develops various software products, each one requiring a software build. A software product is identified by its name and version. A software build identifies the components needed for a particular product. Components may be shared among products. The build status for a product is the lowest status of any of its components. Each component has a status of ready, usable, and not-ready. Information required about each component includes size, programming language (C, C++, C#, Java, or PHP), component name and 3 character version. Each component has one person identified as its owner. People have a unique ID (5 digit number), name (60 characters) and seniority. Components get their status as a result of a software inspection (peer review). An inspection event covers one component and results in an inspection score (0 .. 100). If the score is greater than 90, the component is considered “Ready”. If the score is less than 75, it is “not-ready”; otherwise it is “usable”. Inspection data includes what was inspected, the date of inspection, who conducted the inspection, inspection score, and textual description. The textual description can be updated later, but the score can never be changed. Components may be inspected multiple times.

The actual applications to be run against the database have not yet been determined by the user; however you have been asked to start the development of the database to get it ready. Your assignment is divided into three phases. At the end of each phase, you will be provided with any missing information needed to complete the next phase

Project Phases

1. ER Diagram and Initial Relational Design (20 pts; Due: 10/26):

- a. Construct an ER Diagram with attributes: The ER diagram you create must support all requirements stated above. If you add any restrictions or information not stated above, please explain. Indicate any design requirements that are not included in the ER diagram.



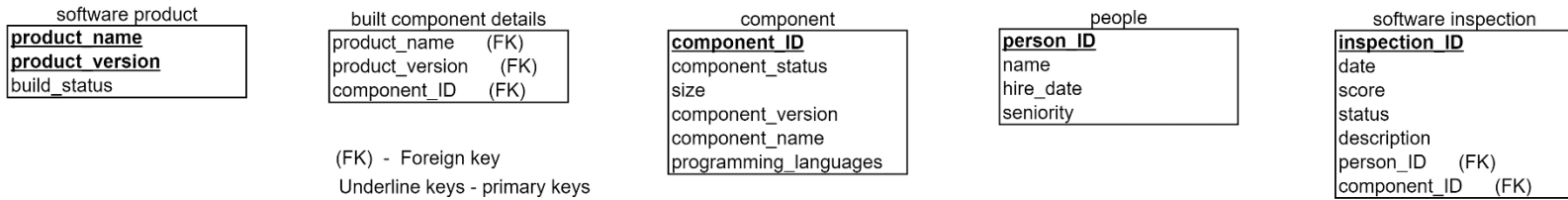
Inspection_ID, Component_ID has been added. hire_date is used to derive seniority. Status is derived from score, component_status is derived from status & product_status is derived from component_status.

- b. Produce an initial Relational design: Given your ER diagram, provide an initial description of your relations, keys and foreign keys.

Underline attributes are primary keys.

Person_id, & component_id in 'software inspection' is foreign key reference to person_id of 'person' & component_id of 'component' respectively. 1 Inspection is done by 1 person.

Built component details indicates all components in a software product built.



c. Submit for grading your ER diagram and relational schemas.

2.a Database Implementation (30 pts; Due: 11/16):

- d. Using SQL DDL statements, create the relations as designed in phase 1. You must include any needed data constraints and keys (primary and foreign) to ensure design requirements are met.
- e. Populate the relations using data provided by the user.
- f. Submit for grading proof of creation of the relations and their population. This could be output of 'Describe' and SELECT * statements. Be sure to indicate your selection of DBMS and location of implementation.

2.b Triggers (10 pts; Due: 11/16):

- g. Select one nontrivial trigger that is needed to ensure data requirements are met.
- h. Show the implementation of that trigger along with the results of your testing to confirm the trigger works as expected in an efficient manner. Continue to implement all required triggers.

3.a Applications (20 pts; Due: 12/6)

- i. You will be provided a list of application requirements
- j. Using the relations populated in Phase 2, you are to create SQL code to implement a set of queries against the database.
- k. Submit for grading, proof of execution of these queries and documentation of any changes made to your implementation.

3.b Final Testing (20 pts; Due: 12/6):

- l. When you have submitted Phase 3.a for grading, Dr. Moore will perform his own testing of the database. Each student will have the same tests performed.

NOTES:

1. Requirements are subject to change at any time at the discretion of the user.
2. If you change details of an earlier phase implementation, please provide detail of this with your submission during the next phase.
3. You will receive a written grade on each phase while you are working on the next phase.