

From Logical to Physical Models

Note: This assignment should be completed as a team. You will be graded later on your contribution to the team.

Background: In this assignment, you will convert your logical DFDs and ERDs to physical models. Start with the logical models that your team submitted at the mid-semester deliverable. Your decisions on how to convert the models will largely be based on your decisions of acquisition strategy and hardware/software. Thus, refer back to your (and your team members') work on previous assignments. If you (or your team) decided to purchase pre-packaged software, you likely wouldn't do physical models in real life, but create them anyway for practice – just list the details of how you think the pre-packaged software would be designed.

Overall formatting and submission instructions: Submit a single PDF file via Canvas before the deadline.

Part 1: Physical Data Flow Diagram

Convert your Level 0 DFD (you don't convert Context Diagrams) to a physical DFD by completing the following steps. You do not have to convert your Level 1 diagrams for this assignment, but you will need to in the final deliverable.

1. Add human-system boundaries in appropriate places on your DFD. (3 pts)
2. Add implementation details to data stores. These will likely be database names. If you are custom developing or customizing, note which type of DBMS you are using (e.g., MySQL, Access, SQL Server). If you are purchasing pre-packaged software, use the name of the software (e.g., "BookingX software: Customer files" (3 pts)
3. Add implementation details to data flows. These could be field names in databases or names of paper forms or digital reports or could be marked simply as data stored in memory being processed by computer code (e.g., "MySQL Customer table attributes" or "Customer report generated on-screen" or "Customer information input through HTML form" or "Customer information in memory being processed by Java code") (3 pts)
4. Add implementation details to processes. These will likely be screens or application modules (e.g., "Input Customer data -- Customer input form created in Access") (3 pts)
5. Add any system-related flows, processes, or data stores that you feel are necessary. Did your team state in nonfunctional requirements that there should be log in screens? Backups created? Logs or copies of deleted data? This step will not be graded in this HW assignment, but in the final deliverable it should be consistent with information you have provided elsewhere in the deliverable.

Part 2: Physical Data Model

Start with the logical E-R diagram, and complete the following steps to convert it to a physical database design.

1. Add a primary key for each table (if you had identifiers in your logical model, it is only a matter of marking it "PK") (3 pts)
2. Create intersection tables for any many-to-many relationships. (Example: If you had a many-to-many relationship between Rooms and Reservations, you should now add a table in between them called Room-Reservation, with a composite primary key.) (3 pts)
3. Add foreign keys to represent all relationships. (See lecture slides or Chs 6 or 11 or do an online search if you don't remember how to do this from ISDS-402) (3 pts)
4. List data types for each field. (Use standard data types, as listed at http://www.w3schools.com/sql/sql_datatypes.asp) (3 pts)
5. Add any necessary system-related tables or fields. (Hint for reservation system project: The project sponsor wants to keep track of cancelled reservations, so the Reservations table should have an attribute to record this.) (2 pts)

2 points for clear wording with no grammatical errors;

2 points for formatting that enables readability and understanding.