Project: Container Terminal Operation

Your company, Apasaja Pte Ltd, has been commissioned to design and implement a database application to manage the container operations at the Persinggahan terminal. The application tracks ships arriving at the different berths and containers being loaded, unloaded, and transferred in the terminal's yard.

The application stores the containers' ISO standard (ISO 6346) identification (e.g., MSCU5285725), and other information left to your design decision. It stores the ships' Maritime Mobile Service Identity (MMSI) number (e.g., 563159100), name (e.g., PACIFIC NINGBO), flag (e.g. Singapore), dimensions (e.g., length 129.57 meters and her width is 20.83 meters), and other information and other information left to your design decision. It stores the berths' identifier (a natural number) and other information left to your design decision. The application stores the locations in the yard. The terminal has a single yard. A location in the yard is identified by its bay, row, and tier numbers (natural numbers identifying the bay, the row, and the tier.) Use natural primary keys. Do not invent identifiers.

The application tracks the real-time and expected status of containers and ships. It tracks the ships' actual and expected arrival and departure times and assigns (available) berths. It tracks container movements in the yard, including loading from a location to a ship at berth, unloading from a ship at berth to a location, and transfers between locations within the yard. We assume that the past, current, and expected arrival, departure, start and end dates and times of each operation are always known (although the expected date and times can be updated, if necessary). The application conserves historical data about ships' schedules (arrival and departure dates and times) and containers' movements (start and end dates and times) but does not track updates on these dates and times.

You are allowed to revisit the simplified requirements above to make them more realistic. If you wish to do so, consult with the teaching team.

1. (4 points) Conceptual Design

   Draw the entity-relationship diagram for the application, including candidate keys and cardinality constraints, in the notations of the lecture.

2. (4 points) Logical Design

   Write the SQL data definition language code (`CREATE TABLE`) with the integrity constraints resulting from translating the entity-relationship diagram.

3. (3 points) Data

   Populate the database with fake but realistic data.

4. (4 points) Workload

   Prepare a workload (a set of SQL data manipulations to demonstrate the database's capability). For instance, schedule a ship for arrival and departure and assign it an available berth, log a container's movement, write views that extend and add their statuses to the tables for locations (available, occupied), containers (in a terminal, not in the terminal), and ships (at berth, at sea and loading, unloading, etc.), and generate reports of daily movements. The workload consists of insertions, deletions, updates, queries, and view creations that involve every table in the schema and illustrate their role in the application. The workload does not need to be exhaustive.

5. (4 points) Checking Constraints with Queries

   There are many constraints on the schedules and movements. For example, time intervals should not overlap, a vessel should be at berth when a container is loaded, the location to which a container is unloaded or transferred should be available, etc. Write queries that check that the schedules and movements are consistent.

6. (3 points) Enforcing Constraints with Triggers

   If possible, use stored functions and triggers to enforce (some of) these constraints. This is the most challenging part of the project. Not every group may be able to attempt.

7. (3 points) Deliverables

   The deliverables are an initial entity-relationship diagram (Question 1) and an initial logical schema (Question 2) in Week 10 and a video presentation and demonstration in Week 12. The video presents the conceptual (Question 1) and logical design (Question 2), the data (Question 3) and workload (Question 4) generation, justifying the design and implementation choices. If applicable, describe your implementation of constraints (Question 5) and triggers (Question 6). The video demonstrates the application features following the proposed workload.

   (a) The entity-relationship diagram is submitted as a PDF image in a `pdf` file.
   (b) The initial logical scheme is presented as the SQL data definition language code (CREATE TABLES with constraints in a `.sql` file)
   (c) The video consists of slides, animations, camera, and screen recordings with voice-over commentary.
   (d) The video must be in MP4 (`.mp4`) or in QuickTime (`.mov`) file format no longer than 15 minutes and no larger than 200MB (you may use `handbrake.fr` to reduce the file size, if needed.)