1. The final exam is cumulative, so you can refer to all the midterm questions to practice, in addition to the question templates below. **IMPORTANT NOTE:** Needless to say, the questions on the actual final exam might differ from the ones in this skeleton. This skeleton is just meant to help you practice, not to be a specification of the actual final exam.

2. In at most two sentences each, explain the meaning of the following terms as they relate to database systems:

   (a) Transaction Consistency.

   ...

3. Given a statement about functional dependencies, Boyce-Codd Normal Form, and/or Third Normal Form, decide whether it is true or false. If you decide it is true, provide a proof. If you decide it is false, provide a concrete counterexample (i.e., a relation instance with its associated functional dependencies) for which the statement doesn’t hold. An example of such a statement that you might be given is: “Every relation \( R(A, B) \) with exactly two attributes is **always** in Boyce-Codd Normal Form, regardless of the functional dependencies that hold over the relation.”

4. Consider a relation \( R \) with attributes \( A, B, C, \) and \( D \):

   (a) Consider each of the following hypothetical “rules” or “axioms” about functional dependencies, and either show (1) that it always holds by giving a proof or (2) that it is false by giving a concrete counterexample, which should be a specific instance of \( R \) that violates the statement. If you decide to produce a counterexample, the instance of \( R \) that you give should consist of **exactly two tuples**. **We will not give you credit for longer counterexamples.**

      - If \( AB \to C \) and \( A \to D \), then \( C \to D \).

      - ...

   (b) Suppose that the following set \( F \) of two functional dependencies holds over \( R \):

      \[ F = \{ A \to B; B \to C \} \]

      Is the decomposition of the schema of \( R \) into \( R_1(A, B, C) \) and \( R_2(A, D) \) a lossless join decomposition? Circle one: **YES NO**. Justify your answer.

   (c) Consider a relation \( S \) with attributes \( A, B, C, \) and \( D \). Write an SQL query that returns an empty answer if and only if the functional dependency \( AB \to C \) holds on relation \( S \). (It is not important what your query returns when the functional dependency does **not** hold on \( S \), as long as the query result is not empty in this case.) Assume that no NULL values are present.
5. Consider the following (simplified) portion of the Dinky schema from the textbook, and which we studied in class:

- CREATE TYPE theater_t AS ROW(tno INTEGER, name TEXT, address TEXT, phone TEXT) REF IS SYSTEM GENERATED;
- CREATE TABLE Theaters OF theater_t REF IS tid SYSTEM GENERATED;
- CREATE TABLE Nowshowing (film INTEGER, theater REF(theater_t) SCOPE Theaters, start DATE, end DATE);
- CREATE TABLE Films (filmno INTEGER, title TEXT, stars VARCHAR(25) ARRAY [10], director TEXT, budget FLOAT);

Assume that the filmno attribute is the primary key of table Films, and that the film attribute of table Nowshowing is a foreign key into table Films. Write a SQL query, using object-relational features as appropriate, that returns ... (The query that you might be asked to write might involve the COLLECT aggregate operator and other multiset and object-relational features that we discussed in class.) **Note: You should use the (standard) object-relational SQL syntax that we studied in class.**

6. One or two questions on query optimization and transaction processing along the lines of the examples and discussion that we went over in class.