1. Let \( q \) have scheme \( ACD \), \( r \) have scheme \( AB \), \( s \) have scheme \( BCD \), and \( t \) have scheme \( BC \).

(a) (7 points) Use the rules of query optimization to rewrite the following query in an optimal form. (Push \( \pi \) and \( \sigma \) over \( \bowtie \) and \( \cup \) as far as possible.)

\[
\sigma_{A=1} \pi_{AC}((r \bowtie s) \cup (q \bowtie t))
\]

(b) Assume that on the average a seek (head move) takes 8 milliseconds, rotational latency (spin under) takes 4 milliseconds, a block transfer takes 68 microseconds. Assume that there are 32,000 records in \( r \) and 128,000 records in \( q \). A block has 512 bytes. All attributes are 4-byte integers. The key for \( q \) is \( A \), for \( r \) is \( AC \), for \( s \) is \( C \), and for \( t \) is \( BC \).

i. (3 points) Estimate the access time for \( \sigma_{B=1} r \) if \( r \) is stored contiguously as a sequential file. (A formula with numbers plugged in is sufficient; you need not do the arithmetic.)

ii. (3 points) Estimate the access time for \( \sigma_{C=1} r \) if \( r \) is a sequential file stored in random blocks scattered on the disk. (A formula with numbers plugged in is sufficient; you need not do the arithmetic.)

iii. (5 points) Estimate the maximum access time for \( \sigma_{A=1} q \) if \( q \) has a \( B^+ \)-tree index for attribute \( A \) with only its root node in memory and all other nodes on disk. Assume that a \( B^+ \)-tree pointer occupies 4 bytes and that the size of a \( B^+ \)-tree node is the size of a block. (For this problem, an exact number to the nearest millisecond is expected.)
2. (15 points) Create an OSM diagram for the following.

An ATM Machine has a Serial# and Company and Location; a Serial# uniquely identifies an ATM Machine as does the machine’s Company and Location together. Some ATM Machines operate for only a limited time, some have upper-limit withdrawal allowances, and some have both limits on time and upper-limits on withdrawals. ATM Machines that operate for only a limited time are called Limited-Time ATM Machines, and ATM Machines that have upper-limits on withdrawals are called Upper-Limit ATM Machines. Limited-Time ATM Machines have a Start Time and a Stop Time. Upper-Limit ATM Machines have a Dollar-Amount Limit, which is currently set at $1,000.

In your diagram, use object set names (noun phrases that start with capital letters) as given in the description. Also, be sure to name all relationship sets and give proper participation and co-occurrence constraints. Make appropriate use of generalization/specialization. You must also give a formal predicate-calculus statement with quantifiers to capture the $1,000 constraint.
3. Translate the following OSM application model into the set of predicates and rules that characterize it as specified below.

(a) (3 points) Give the object-set predicates.

(b) (3 points) Give the relationship-set predicates.

(c) (5 points) Give the referential integrity rules.

(d) (5 points) Give the generalization/specialization rules.

(e) (5 points) Give the participation-constraint rules for the relationship set named \textit{A B C relationship}.
4. (13 points) Convert the following ORM diagram into an ORM hypergraph. Make your ORM hypergraph correspond to the results that would be obtained by the hypergraph generation algorithm (Algorithm 9.1).
5. (15 points) Reduce the following ORM hypergraph to a canonical hypergraph. Assume that semantic equivalence holds where you need it. You may do your reduction by clearly marking components to be deleted, by redirecting edges, by adding roles to make connections congruent, and by circling equivalence classes that should be lexicalized.
6. Consider the following ORM hypergraph.

(a) (8 points) Give the set of schemes produced by the scheme synthesis algorithm (Algorithm 10.1). Also, add scheme names $R_1$, $R_2$, ... for the schemes.

(b) (4 points) Underline the keys in your schemes above.

(c) (6 points) Give the inclusion dependencies as produced by Algorithm 10.2. Use $r_1$, $r_2$, ... to designate the relations associated respectively with $R_1$, $R_2$, ...
7. Consider the following generic relational database scheme.

\[
\begin{align*}
\text{Person} & : (ID, Name, Address, DLNumber, PhoneNr) \\
& \quad \text{Primary Key: ID} \\
& \quad \text{Key: Name, Address} \\
& \quad \text{Key: DLNumber} \\
\text{Vehicle} & : (LicensePlateNr, Price, Color) \\
& \quad \text{Primary Key: License Plate Nr} \\
\text{Owns} & : (ID, LicensePlateNr) \\
& \quad \text{Primary Key: ID, License Plate Nr} \\
& \quad \text{Owns[ID] \subseteq Person[ID]} \\
& \quad \text{Owns[License Plate Nr] \subseteq Vehicle[License Plate Nr]}
\end{align*}
\]

(a) (7 points) Give the SQL create statement for the Owns relation scheme. Declare appropriate types for each attribute.

(b) (6 points) Write an SQL query for this scheme to list the ID and Name of car owners who own vehicles costing less than $3,000.
8. For each pair of sets of FDs $F$ and $G$ below determine whether $F \equiv G$. If so, show how to derive each FD of $G$ not already in $F$ from $F$ and each FD of $F$ not already in $G$ from $G$. (You can use the closure of a set of attributes for this if you wish.) If $F$ is not equivalent to $G$, give all FDs of $G$ that cannot be derived from $F$ (if any) and all FDs of $F$ that cannot be derived from $G$ (if any).

(a) (7 points) $F = \{A \rightarrow B, CD \rightarrow A, C \rightarrow D\}$ and $G = \{C \rightarrow B, A \rightarrow B, C \rightarrow AD\}$

(b) (7 points) $F = \{A \rightarrow B, AB \rightarrow C, B \rightarrow CD, A \rightarrow C\}$ and $G = \{A \rightarrow BC, B \rightarrow D, C \rightarrow A\}$
9. Let $F = \{ A \rightarrow BC, B \rightarrow A, BC \rightarrow E, AD \rightarrow C, E \rightarrow I, G \rightarrow EI, GE \rightarrow H \}$.

(a) (3 points) Produce the ORM hypergraph corresponding to $F$. Each FD should correspond to one edge.

(b) (12 points) Give the ORM hypergraph after making head and tail reductions.
10. Let $U = ABCDE$ and let $F = \{AB \rightarrow C, C \rightarrow E, E \rightarrow AD, A \rightarrow D\}$.

(a) (5 points) List all candidate keys for $ABCDE$.

(b) (10 points) Decompose $ABCDE$ into BCNF. (Hint: consider using a hypergraph.)

(c) (2 points) Is it possible to have a database scheme for $U$ that is both in BCNF and dependency preserving?
11. Let $R = ABCDEGH$ and let $F = \{ ABC \rightarrow D, D \rightarrow E, E \rightarrow AB, E \rightarrow G, G \rightarrow H \}$ be a set of FDs over $R$.

(a) (2 points) Give the hypergraph for $F$. Each FD should correspond to one edge.

(b) (3 points) Identify all properly embedded FDs.

(c) (7 points) Reduce the hypergraph so that it is canonical. Be sure that your reduction preserves information and constraints. (Assume semantic equivalence.)

(d) (10 points) Use scheme synthesis (Algorithm 10.1) to produce schemes. Underline the keys in each scheme.

(e) (3 points) Which schemes must the implementation join together to check the FD $ABC \rightarrow D$?
12. Consider the following canonical hypergraph.

(a) (2 points) Which object set is included in the most functional closures (of other object sets)?

(b) (7 points) Beginning with Grad Student, apply the nested scheme generation algorithm (Algorithm 10.3) to the hypergraph. For any succeeding tree, select the root to be the (or any one of the) not-already-marked object sets included in the most functional closures.

(c) (7 points) Beginning with Department, apply the nested scheme generation algorithm (Algorithm 10.3 in the text) to the hypergraph. For any succeeding tree, select the root to be the (or any one of the) not-already-marked object sets included in the most functional closures.