INSTRUCTIONS:

(1) Print your **name and student ID** at the top of all pages and on the cover page.

(2) Let the proctors know immediately if you are missing any exam pages.

(3) Please do not separate the pages of this exam.

(4) Write your answers legibly in the space provided, and **show your work**. If you need more space, you may use backs of pages, but indicate clearly that you are doing so.

(5) If required for a given question, or if in doubt, clearly **state any reasonable assumptions** and then proceed to complete the question.

(6) Grading will take into account the **correctness and clarity** of your solutions, as well as the **soundness of any assumptions** you make. Course staff will not comment on the correctness, clarity, or soundness of your solutions during the exam.

(7) Read an entire question before you begin to answer it. Questions vary in difficulty and weight, so **try not to get stuck** on harder questions.

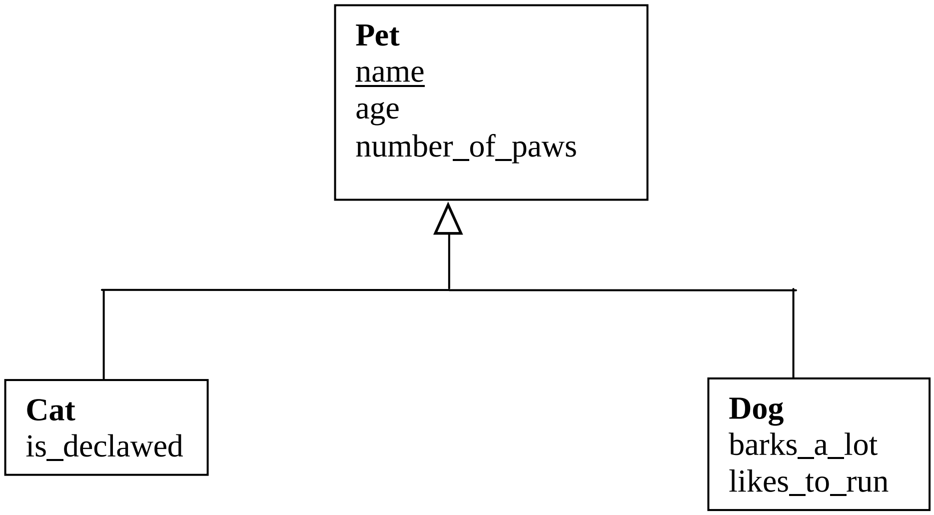
Good luck!

The five minute rule (Gray and Putzolu, 1985):

Pages referenced every five minutes should be memory resident.

Question 1: Entity-relationship modeling [5 marks]

Consider the following E-R model of a pet database:



Part (a): [3 marks]
In lecture, two different techniques were taught for translating such E-R models to relational schemas. Apply each of these techniques to the above model, indicating clearly all primary keys and foreign key constraints. Explain in one sentence the main advantage of each technique.

Part (b): [2 marks]
How does the model change if we would like to represent a "likes" relationship between pairs of pets such that each pet is required to like at least one other pet? Answer by drawing any necessary modifications directly on the diagram above. You do not need to repeat part (a) for the modified diagram.

Question 2: Query languages [5 marks]

Consider the following relational schema used to describe delivery of parts by suppliers to projects (primary keys are underlined):

Supplier (<u>sup_num</u>, sup_name, city)
Part (<u>part_num</u>, part_name, colour, weight, city)
Project (<u>proj_num</u>, proj_name, city)
SPP (<u>sup_num</u>, <u>part_num</u>, <u>proj_num</u>, qty)

Explanation:

- Supplier defines the supplier number, name and the city in which the supplier is located.
- Part defines the part number, name, colour, weight, and the city where the part is stored.
- Project defines the project number, name, and the city in which the project is located.
- SPP records deliveries of a part from a supplier to a project, including the quantity.
- The attributes sup_num, part_num, and proj_num in SPP are foreign keys that reference the primary keys of Supplier, Part and Project, respectively.

Answer the following queries using the indicated query languages.

Part (a) [1 mark]
Using relational algebra, find the <u>names</u> of suppliers who have delivered at least one part to at least one project located in Waterloo or London.

Part (b) [1 mark]
Using relational algebra, find the <u>supplier numbers</u> of suppliers who have delivered parts to at least two projects. (Hint: it is not necessary to use aggregation to answer this question.)

Part (c) [1 mark]

Using SQL, find the number of distinct suppliers who have delivered part number 105 to some project. (The part may be delivered to different projects by different suppliers.)

Part (d) [1 mark]

Using SQL, find the part numbers of all parts that have never been delivered to any project.

Part (e) [1 mark]

Using SQL, find the names of suppliers and for each supplier the total quantity of parts that supplier has ever delivered to any project. Sort the output in descending order by supplier name.

Question 3: Constraints and database design [4 marks]

The following questions pertain to the schema from Question 2. Assume that no indices exist initially.

Part (a) [1 mark]
What referential action (associated with a foreign key constraint) would ensure that a project cannot be deleted from the Project table if the SPP table contains one or more records referencing that project?

Part (b) [3 marks]
For each of the following questions, determine what index or indices would be best for solving the stated problem. For each index, specify the table, the search key, and the index type (B^+-tree vs. hash). In each case, use as few indices as possible.
Assumptions: all data (i.e., tables and index structures) are stored on a single hard disk, all tables are stored in heap files, and the inner nodes of all B^+-tree indices are cached in memory.

(i) [1 mark]
When a row is inserted into the SPP table, we wish to check that this row references an existing row in Supplier. (I.e., we wish to enforce a foreign key constraint.)

(ii) [1 mark]
Given a supplier number and project number, we wish to determine the part numbers of all parts delivered by that supplier to that project.

(iii) [1 mark]
Given a letter of the alphabet, we wish to find all the suppliers located in cities starting with that letter (e.g., given letter "W" the query returns Waterloo, Windsor, etc.).

Question 4: Functional dependencies [4 marks]

Let R denote a relation schema with four attributes: (A,B,C,D). Assume that R is in 1NF. Consider the following set of functional dependencies over the attributes of R:

$F = \{ \quad D \rightarrow B, \quad B \rightarrow C, \quad AD \rightarrow C \quad \}$

Part (a) [3 marks]
Prove using Armstrong's Axioms (and only those axioms) that $AD \rightarrow BC$ follows from F. Show all the steps of your proof and for each step state clearly which axiom you are applying.
Hint: Armstrong's Axioms are *reflexivity*, *augmentation*, and *transitivity*.

Part (b) [1 mark]
Determine whether A is extraneous with respect to F on the left side of $AD \rightarrow C$. Explain your answer.

Question 5: Normalization [6 marks]

Let R denote a relation schema with four attributes: (A,B,C,D). Assume that R is in 1NF.

Part (a) [3 marks]

Consider the following set F of functional dependencies over the attributes of R:

$F = \{ \quad AB \rightarrow CD, \quad AC \rightarrow B, \quad B \rightarrow A \quad \}$

Determine whether R is in 3NF with respect to F. If not, give a 3NF decomposition of R using the algorithm taught in lecture. Show your work in detail and state your final answer clearly.

Part (b) [3 marks]

Consider the following set G of functional dependencies over the attributes of R:

$G = \{ \quad A \rightarrow B, \quad C \rightarrow D \quad \}$

Determine whether R is in BCNF with respect to G. If not, give a BCNF decomposition of R using the algorithm taught in lecture. Show your work in detail and state your final answer clearly.

Question 6: Storage and file structures [4 marks]

A RAID system is being constructed using ten identical 1TB hard disks. The raw storage capacity of this system is exactly 10 TB. The usable storage capacity is defined as the raw storage capacity minus the storage overhead due to mirroring and parity, and is less than 10 TB.

Part (a) [1 mark]
What is the usable storage capacity if RAID 10 (mirrored disks with block striping) is used?

Part (b) [1 mark]
Suppose that we want to write a single 150 GB file to the RAID system. Assume the following:

- there is no other I/O activity that might interfere with this write operation
- the maximum write speed of each disk individually is 150 MB/s
- the block size is much smaller than the size of the file under consideration

Estimate how many seconds it takes to write the file if RAID 5 (block-interleaved distributed parity) is used.

Part (c) [2 marks]
Consider the following relational schema:

Pet(pet_id, name, age)
Owner(owner_id, name, address)
Owns(pet_id, owner_id)

Describe (e.g., using a drawing) a multi-table clustering file structure that would be useful for computing the natural join of Owner with Owns. Also explain in one sentence why this structure is advantageous compared to storing each table in a separate file.

Question 7: Indexing and hashing [4 marks]

Consider a table with schema Pet(name, species, age), and a hash index on the species attribute. The hash function *h* is defined over strings, and its values (in binary) are shown in the table below. The index records entries of the form (species, name) using extendable hashing.

Species	*h*(species)
Dog	001 101 100
Cat	010 111 010
Parrot	011 010 110
Mouse	100 111 001
Pig	111 000 111

Assumptions:

- (species, name) entries are fixed-length
- each bucket can store up to two such entries
- the hash prefix is calculated using the most significant bits, which are leftmost in the table

Part (a) [3 marks]
Assuming the index is initially empty, draw the state of the extendable hash structure after the following entries are inserted:

(cat, Fluffy), (parrot, Polly), (cat, Snowy), (pig, Peppa), (mouse, Mickey)

In your drawing be sure to indicate the length of the hash prefix both for the bucket address table and for each bucket.

Part (b) [1 mark]
If the Pet table is stored as a sequential file sorted by name, then is a secondary index on species considered sparse or dense?

Question 8: Transactions and concurrency control [9 marks]

Consider the transaction schedule shown below. (Time increases in the downward direction.)

T_1	T_2	T_3
read(A)		
	read(B)	
write(B)		
	read(A)	
write(A)		
		read(B)
		commit
	commit	
commit		

Part (a) [2 marks]
Is this schedule conflict-serializable? Explain using a precedence graph.

Part (b) [1 mark]
Is the schedule recoverable? Explain why or why not.

Part (c) [3 marks]

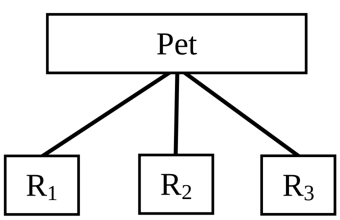
Is the schedule possible using two-phase locking? If so, rewrite the schedule by adding lock/unlock instructions for each transaction. If not, explain why not. Follow these rules: (i) you can only use S and X locks; (ii) a transaction may upgrade an S lock directly to an X lock without releasing it first; (iii) you cannot reorder the read and write instructions; (iv) you need not follow strict or rigorous 2PL.

Part (d) [1 mark]

Suppose that transaction T_4 is younger than transaction T_5. Suppose that T_5 holds an exclusive lock on an object C, and then T_4 requests a shared lock on the same object. What happens in this situation if the "wound-wait" strategy is used for deadlock prevention?

Part (e) [2 marks]

In multiple granularity locking, suppose that two levels of locking are defined: table and record. Suppose that initially no transaction holds any locks. Next, transaction T_6 reads record R_2 of the Pet table in the diagram below, and transaction T_7 writes record R_1. What locks do T_6 and T_7 request? Do any of the locks needed by T_6 conflict with any of the locks needed by T_7?



Question 9: Recovery [5 marks]

Suppose that the database recovery log contains the following records just before a system crash:

Record #	**Log record**	
1	<T_{10}, start>	**beginning of log**
2	<T_{10}, B, 100, 150>	
3	<T_{11}, start>	
4	<checkpoint {T_{10}, T_{11}}>	
5	<T_{11}, C, 200, 250>	
6	<T_{10}, B, 100>	
7	<T_{10}, abort>	
8	<T_{12}, start>	
9	<T_{12}, A, 500, 400>	
10	<T_{11}, C, 250, 251>	
		end of log just before system crash

Assume that ordinary checkpointing (as opposed to fuzzy checkpointing) is used.

Part (a) [3 marks]
If the recovery algorithm presented in lecture is used, what log records are appended to the log during recovery?

Part (b) [1 mark]
What is the record # of the earliest log record visited during the undo phase of recovery? (Earliest means smallest record #.)

Part (c) [1 mark]
What is the final value of object C after recovery?

Question 10: Key-value storage [4 marks]

A small Canadian start-up company is setting up a Cassandra cluster distributed across three datacenters: one near Vancouver, one near Toronto, and one near St. John's. Every datacenter is connected directly to every other datacenter using a high-speed optical network. Each data object is replicated three ways, with one copy in each data center.

Part (a) [3 marks]

Suppose that a client in the Vancouver datacenter issues a Get request using QUORUM consistency. Describe in detail (e.g., using a detailed drawing) the flow of protocol messages used to process this request. How many acknowledgments does the coordinator wait for? Which replicas are likely to respond first? <u>Assumptions:</u> no failures, all three replicas of the object have the latest value.

Part (b) [1 mark]

What is the rationale for using de-normalized schemas when working with key-value storage systems? Answer in one sentence.