Sample Questions for the Last Part of the Final Exam

In this last part of the final exam you will be given one or more questions that will require a free-form "essay" style answer, between ½ and 1 page long. You may potentially be given a choice between several questions. Here are some *examples* of the questions that you might see. *The exam may include questions not listed below.*

What is "the relational model" and what are its main strengths and weaknesses?

What is "normalization" and what do we need it for?

What may be some of the advantages of *not* normalizing a table?

What is a "storage engine" in MySQL and why is there a need or more than one?

What is the "SQL injection" attack and how can it be prevented?

What are the pros and cons of creating an index on a field?

What does XML have to do with relational databases?

What is the difference between "an ANSI join" and "a traditional join"? Which one is better and why?

What is the difference between the ER model and the relational model?

What is a "Cartesian product" in the context of SQL joins and how do we avoid it?

What is a "natural join" and why should one avoid it?

Suppose that you learn that your client has a database table with 50 fields. Should this be a source of concern? Why or why not?

Harrington's *Relational Database Design* says that 2NF requires that "[a]ll non-key attributes are functionally dependent on the entire primary key. " (p. 111). In this case, what are "non-key attributes"? Illustrate your answer with an example.

What are the benefits of using foreign key constraints in a relational database?

What is a "candidate key"? How is it different from a "primary key"? Provide an example.

MySQL's "InnoDB" storage engine enforces foreign key constraints. Some of the other storage engines ignore them. What could be the advantages of using a storage engine that does **not** support foreign key constraints?

Can a table have a foreign key to its own primary key? If not, explain why not. If yes, explain

what this would achieve and whether this foreign key can be NOT NULL (and why).

What are “natural” primary keys and what are pros and cons of using them?

What's the difference between an “inner join” and an “outer join”? When would you want to use an outer join?

How can we use views to control access to data?

Harrington's *Relational Database Design* says that 2NF requires that “[a]ll non-key attributes are functionally dependent on the entire primary key” (p. 111). What does she mean by “functionally dependent”? Provide an example.

Harrington's *Relational Database Design* says: “True one-to-one relationships are very rare in business ” (p. 66). Explain why this is the case.

While reviewing an ER diagram drawn by an inexperienced data modeller you notice that entities called “User” and “Book” are connected with an arrow that says “searches for.” Why is this likely to be a mistake? What would be a possible *good* reason to have this relationship?

What is an “insertion anomaly”? Provide an example.

Why is it a bad idea to store user's passwords in your database? What shall you do instead?

Suppose that a table called “student” has fields “student_no” (the primary key), “utorid” and “last_name”. The fields “student_no” and “utorid” functionally determine each other. They both also functionally determine “last_name”. Is this a 3NF violation? Explain.

You are creating a table to store information about movies. What data type would you choose for the movie title? Explain the advantages and disadvantages of your choice (provide examples).

You are creating a table to store information about paintings in an art gallery. How would you represent the time when the painting was produced? Explain the advantages and disadvantages of your choice (providing examples).

Suppose that the City of Toronto has a database of parking spots in the city which has up-to-date information about fees and the hours during which parking is restricted. They want to make this information available in real-time to companies that would want to incorporate them into services such as Google Maps. How can they do this? Provide specific examples.

What are “NoSQL” databases and what are their advantages and disadvantages relative to SQL databases?

What is the difference between a “correlated” and an “uncorrelated” subquery?

Can we store a year in a field defined as char(4)? (If so, why do we need the “year” datatype then?)

Suppose a database has an entity called “purchase_order” to which multiple “order_items” are linked by means of order_item having a FK to purchase_order. Each order_item has an attribute “price.” Would it make sense for purchase_order to also have an attribute “total_price” to record the total amount the customer was charged? What would be the advantages and disadvantages of doing so?

Suppose a query has a clause that says “where count(*) > 100”? Can this be correct? If so, explain what this clause would achieve. If not, explain what the author of the query probably should have done.

Suppose you want to record information on 1,000,000 customers storing for each customer the following information: customer_id, name, email address, phone number. Roughly how much storage space (in bytes, kilobytes, megabytes, gigabytes, or terabytes, which ever is most appropriate) will be likely be needed for this? Explain how you arrived at your estimate.