NAME (PRINT):

Last/Surname First /Given Name

STUDENT #: **SIGNATURE:**

UNIVERSITY OF TORONTO
FACULTY OF INFORMATION
INF1343: Data Modeling and Database Design
SAMPLE FINAL EXAM
Duration - 2 hours 50 minutes
Permitted Aids: None

No aids are permitted in this exam. You are reminded that you may be charged with an academic offence for possessing any unauthorized aids during the writing of an exam, including but not limited to any electronic devices with storage, such as cell phones, pagers, personal digital assistants (PDAs), iPods, and MP3 players. Unauthorized calculators and notes are also not permitted. Do not have any of these items in your possession in the area of your desk. Please turn the electronics off and put all unauthorized aids with your belongings at the front of the room before the examination begins. If any of these items are kept with you during the writing of your exam, you may be charged with an academic offence. A typical penalty may cause you to fail the course.

This exam is worth 35% of your grade and consists of 350 points.

Most of the questions on this exam refer to the case of Up! — a hypothetical small company that services elevators in the Greater Toronto Area. Clients who have buildings with elevators hire Up! to do maintenance on their elevators as well as to fix them when problems occurs. Up! has a database of elevators that it is currently servicing. It also has a website that customers can use to report problems and check maintenance status. The next page describes four of the tables in Up!'s database. Three of the tables are described in a "data dictionary" format, while the fourth table is described with the SQL statement that was used to create it. Please examine the table descriptions before answering any questions. Please note that there are other tables in Up!'s database. In particular, there is a table **`employee`** that describes the employees of Up! and a table **`model`** that describes the models of elevators (e.g., model X2000 by United Elevators). You are not provided with a full description of those tables, but you should find all the relevant information in the description of other tables.

Some of the tables in Up!'s Database

elevator

Field	Type	Null	Key
elevator_id	integer	NO	primary key
building_id	integer	NO	references building(building_id)
building_type	enum('office', 'residential')	YES	
model_id	integer	YES	references model(model_id)
last_inspection_date	date	YES	

building

Field	Type	Null	Key
building_id	integer	NO	primary key
client_id	integer	NO	references client(client_id)
address	varchar(100)	NO	
city	varchar(100)	NO	

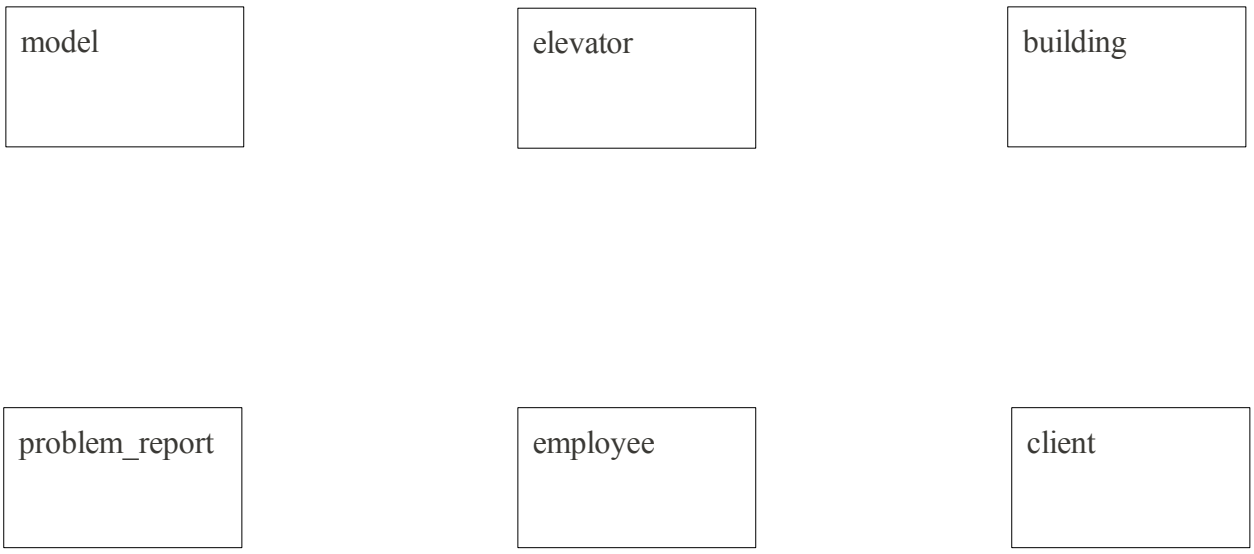
client

Field	Type	Null	Key
client_id	integer	NO	primary key
name	char(100)	NO	
phone_number	char(10)	NO	
notes	varchar(500)	YES	

problem_report

```
create table problem_report (
 problem_id integer not null auto_increment,
 elevator_id integer not null,
 foreign key (elevator_id) references elevator(elevator_id),
 description varchar(500),
 contact_name varchar(100),
 contact_phone_number char(10),
 time_created timestamp default current_timestamp,
 time_fixed datetime,
 status enum('new', 'assigned', 'resolved') not null default 'new',
 assigned_to integer,
 foreign key (assigned_to) references employee(employee_id),
 comments varchar(500),
 primary key (problem_id)
) engine=InnoDB;
```

1. Below are six of the entities in the ER diagram that was used to design Up!'s database. Complete the diagram by drawing the **five** relationships that are implied by the primary and foreign keys in the table descriptions. Do not draw any other relationships – points will be subtracted for unnecessary relationships. Use crow's feet notation and make sure to mark cardinality. Indicate whether the relationship is mandatory or optional if this can be determined. **Label the relationships with what you believe they represent.** You do *not* need to fill in the attributes for the entities. (75 points.)



2. Do any of Up!'s tables violate 2NF or 3NF? If so, indicate the specific functional dependencies that cause the violation and show how to bring the tables to 3NF. If all tables do satisfy 3NF, describe what a 2NF and 3NF violations would look like in the context of Up!'s database. Write your answer in the space below. (35 points)

(Your answer probably doesn't need to go past this point.)

3. Write SQL queries to perform the listed tasks. (30 points per query, 120 points total).

a) Get a count of “new” problem reports.

b) Get a list of open problem reports with corresponding building addresses, sorted by the time when the problem was reported in reverse chronological order (most recent first).

c) List the addresses for all buildings for “Bank of Mississauga” that have 2 or more elevators.

e) Create a new table, “performance_review” which will store information about annual performance reviews for Up!'s employees. The text of a review should be stored in a field called “review_text”. The table should have a mandatory foreign key to the employee who performed the review. It should also record the time when the review was entered.

4. In this last part you will be given three questions, jointly worth 120 points. You will have between ½ and 1 page to answer each question. (Shorter answers are acceptable, as long as they answer the question fully.) 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 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?

Systems like Django allow software developers to write database-driven web applications without SQL. How do they do it and what's the advantage of it?

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.