COMS 4111

Homework 2A

Nathan Gong

1. Briefly explain the concepts of "top-down" and "bottom-up" data modeling.

In top-down data modeling, lower-level entity sets that have distinctive attributes or relationships are specialized from a higher-level entity set. In bottom-up data modeling, lower-level entity sets that share the same attributes or relationships are generalized into a higher-level entity set.

2. What are two reasons it is necessary to use an associative entity to implement a relationship set/relationship between entities?

One reason it is necessary to use an associative entity is if the relationship being implemented is a many-to-many relationship. Another reason to use an associative entity is if there are properties on the relationship that are not attributes of the connected entities.

3. Briefly explain the concepts of the degree of a relationship/relationship set, and the cardinality of a relationship.

The degree of a relationship is the number of entity sets that the relationship involves. The cardinality of a relationship is the number of entities to which another entity can be associated with via the relationship.

4. A theta-join is of the form $r \bowtie_\theta s$, where r and s are relations (tables). What is an equivalent relational algebra statement that only uses $\pi, \sigma, \times, \cup, -, \rho$? Your answer may not need to use all the operators.

$r \bowtie_\theta s = \sigma_\theta(r \times s)$

5. Consider a domain for an attribute/column in which the possible values are in the range 0.00 to 99.99, and there can only be two digits after the decimal point. What SQL data type would you use?

NUMERIC(4, 2)

6. Consider the create table statement below, which columns could be the primary key in a MySQL database and why?

```
CREATE TABLE IF NOT EXISTS course_management.students_fixed (
    CU_id INT NOT NULL,
    uni VARCHAR(12) NULL,
    email TEXT NOT NULL,
    first_name TEXT NULL,
    last_name TEXT NULL,
    middle_name TEXT NULL
);
```

The CU_id column could be the primary key in a MySQL database because it is a unique identifier for a student, and it is marked as not able to be null. The uni column is a unique identifier for a student, but it is able to be null, so it could not be the primary key. The email, first_name, last_name, and middle_name columns are not guaranteed to be unique identifiers for a student, so they could not be the primary key.

7. Why does using a natural join sometimes produce incorrect results or results that do not make any sense?

A natural join automatically joins tables based on columns with the same name. If two tables have columns with the same name but with different meanings, using a natural join might produce incorrect or nonsensical results. Similarly, using a natural join might inadvertently include columns with the same name that were not intended to be used for the join, which might lead to incorrect data associations since natural joins do not allow you to specify which columns to join on.

8. Some SQL database management systems do not implement the full outer join operation. Describe how to write an equivalent query to a full outer join using other SQL capabilities.

An equivalent query to a full outer join can be achieved by performing a left join, performing a right join, then combining the results of the left and right joins with a union.

9. SELECT statements and UNION statements behave differently with respect to duplicates in a result set. Briefly explain the differences. How can you make SELECT and UNION behave the same with respect to duplicates?

SELECT statements return all selected rows, including duplicates by default. UNION statements combine the results from multiple SELECT statements and automatically remove duplicates from the result. To make a SELECT statement remove duplicates, the DISTINCT keyword can be added to the statement. To make a UNION statement include duplicates, the UNION ALL operation can be used.

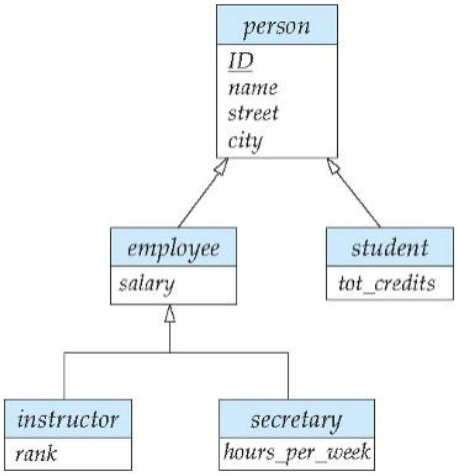
10. Explain the concept of “arity” in SQL statements.

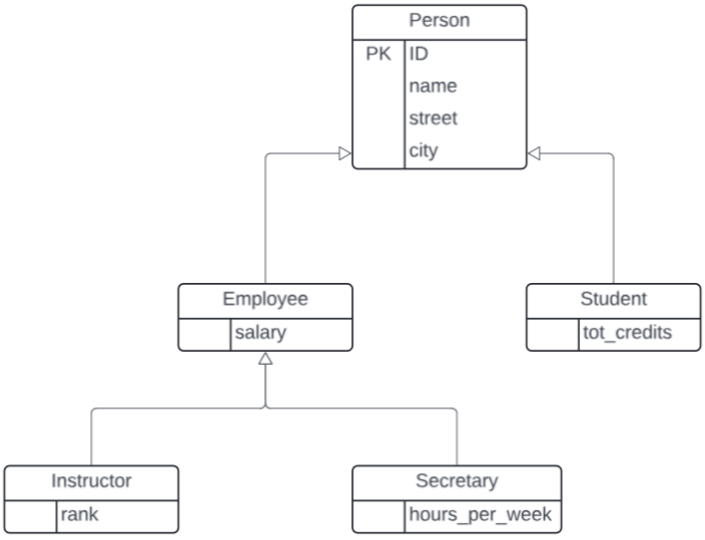
“arity” in SQL statements refers to the number of attributes/columns in a table. This is often relevant when considering set operations, where both input relations in the operation must have the same arity and have compatible attribute domains. Arity also refers to the number of operands that a SQL statement takes, which is important to consider when structuring a SQL function, operator, or statement.

11. Explain the concepts of overlapping, disjoint, complete and incomplete when modeling inheritance/specialization in ER-modeling.

Overlapping/disjoint and complete/incomplete are the two dimensions used to describe constraints involved when modeling ER inheritance/specialization. In an overlapping specialization, an object can be a member of more than one specialized subclass. In a disjoint specialization, an object can only be a member of one specialized subclass. In a complete specialization, every instance of the parent class has one or more unique attributes that are not common to the parent class. In an incomplete specialization, only some instances of the parent class may be specialized, while other instances of the parent class have only the common attributes.

12. Consider the diagram from the lectures slides for our course. Draw the equivalent Crow's Foot Notation diagram. You may have to add/extend Crow's Foot for your diagram. You can use notes or text to explain your extensions.





In the diagram above, Crow's Foot is extended to use separate arrows (e.g. Employee, Student) to represent overlapping generalization and merged arrows (e.g. Instructor, Secretary) to represent disjoint generalization.

13. For the sample data associated with course textbook, the relational expression $course \triangleright prereq$ returns courses without prereqs. a. Provide an equivalent query that does not use anti-join. b. Provide a query that uses anti-join to return courses that are not prereqs. You can just provide the statements. You do not need to execute them.

    a.

π course.course_id, course.title, course.dept_name, course.credits (σ prereq.course_id=null (course ⟕ course.course_id=prereq.course_id prereq))

    b.

course ▷ course.course_id=prereq.prereq_id prereq

14. In the lecture, we discussed three motivations for using views. List and briefly explain the motivations/use cases.

One motivation for using views is to hide certain data from the view of certain users in the case that it is not desirable for all users to see the entire logical model. Another motivation for using views is that it allows for the saving of an expression, which can be easily substituted into queries or used in the definition of another view. Finally, a motivation for using views is that they allow for flexibility and consistency, as the view definition can be modified if the underlying structure of a table changes without impacting its users.

15. What is a materialized view? What is one advantage, and one disadvantage of a materialized view compared to a non-materialized view?

Materialized views are physical copies of relations that are created when the view is defined, which are used in certain database systems that allow view relations to be physically stored. An advantage of materialized views is that they lead to faster performance, as queries no longer needed to be computed at runtime since the data is physically stored. A disadvantage of materialized views is that they must be maintained, as the materialized view will become stale if it is not updated when the underlying relations used in the query are updated.