

F24-W4111: Homework 1A

Overview

- Check CourseWorks and Ed for submission instructions, deadlines, clarifications, etc.
- This homework is a set of written questions testing knowledge and review of concepts covered in:
 - The lectures, both the presentations and any information communicated during the lecture.
 - The slides that accompany the recommended textbook.
 - URL: <https://db-book.com/slides-dir/index.html>
 - You only need to reference the slides for chapters 1, 2, 3, 6, and 7.

Concepts

1. List purposes for/motivations for database systems compared to managing data by writing applications that process files.
2. Database systems provide users with an abstraction view of the data.
 - a. Briefly explain the concept.
 - b. Why do writing applications that access files do not provide an abstraction?
3. What are the 3 levels of data abstraction that a DBMS provides?
4. Explain the difference between database schema and database instance. What two concepts in an object-oriented language correspond to schema and instance?
5. Briefly describe the concepts of data definition language and data manipulation language. What are the two types of data manipulation language?
6. Briefly explain two-tier and three-tier database application architectures. Is a full-stack web application a two-tier architecture or a three-tier architecture?
7. What are the four types of database users based on skill level and for what they use the database. Which type of user defines schema and defines what information users can access?

Relational Model

Understanding Data

The following is data in the SSOL format for our course COMS W4111 but with made up values for Student Name, Student UNI, Student PID, Email, School, Level, Affiliation, Points.

student_name	uni	student_pid	email	school	level	affiliation	points
Ferguson, Donald F	dff9	C001234567	dff9@columbia.edu	CC	U03	CCCOMS	3
Baggins, Frodo	fb001	C001234889	frodo.baggins@shire.gov	EN	U03	ENCOMS	3
Romanoff, Natasha	nr2103	C009999999	nr2103@bamder.com	BC	U03	BCCOMS	3
Prince, Diana	dp2121	C007171717	wonderwoman@avengers.org	BC	U03	BCCOMS	3
Wayne, Bruce	bw9101	C008121212	batman@justice-league.org	GS	U04	GSCOMA	3
Potter, Harry James	hjp1001	C009898989	hjp1001@columbia.edu	BU	P04	BUEXMS	3

Answer the following questions based on your general knowledge of Columbia University and examples given in class.

1. Which attributes are not from an atomic domain?
2. Which attributes are likely to be candidate keys?
3. Which attributes are likely to be foreign keys into another relation?
4. Which attribute is clearly a surrogate key?

An Algebra

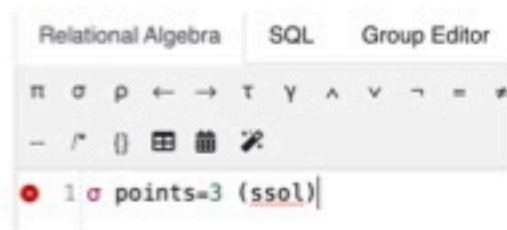
"The result of a relational-algebra operation is a relation and therefore relational-algebra operations can be composed together into a relational-algebra expression."

Assume that the name of the relation above is SSOL. Give two relational algebra expressions that are examples of composition that when applied to the relation above produce the following relation.

student_name	uni	school
Romanoff, Natasha	nr2103	BC
Prince, Diana	dp2121	BC

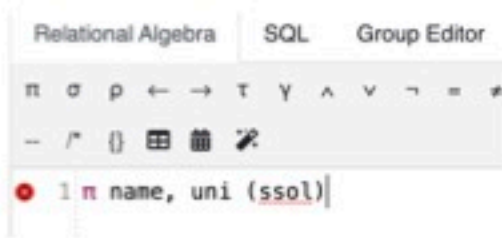
You can use the relax calculator to type your expressions, and then copy/paste the expressions into your answers. The syntax check will indicate errors but you are not executing the expressions.

For example,



shows how to type an expression that is a select which matches all rows in the original relation.

And,



shows how to type an expression that produces

student_name	uni
Ferguson, Donald F	dff9
Baggins, Frodo	fb001
Romanoff, Natasha	nr2103
Prince, Diana	dp2121
Wayne, Bruce	bw9101
Potter, Harry James	hjp1001

Schema

Translate this relation schema definition into an SQL create table statement. This question uses the notation from class slides.

instructor = (UNI, last_name, first_name)

Entity Relationship Modeling

Reverse Engineering

The following slide is from the slides associated with the recommended textbook.



And a Few More Relation Definitions

- **create table** *student* (
 ID **varchar**(5),
 name **varchar**(20) not null,
 dept_name **varchar**(20),
 tot_cred **numeric**(3,0),
 primary key (ID),
 foreign key (dept_name) **references** *department*);
- **create table** *takes* (
 ID **varchar**(5),
 course_id **varchar**(8),
 sec_id **varchar**(8),
 semester **varchar**(6),
 year **numeric**(4,0),
 grade **varchar**(2),
 primary key (ID, course_id, sec_id, semester, year) ,
 foreign key (ID) **references** *student*,
 foreign key (course_id, sec_id, semester, year) **references** *section*);

Use LucidChart to draw an ER diagram in Crow's Foot Notation that is a reverse-engineered, logical model of the schema. You only need to do the entities *student* and *takes*. Do not worry about other referenced entities.

Pros and Cons

List some advantages and disadvantages of ER Modeling.