on2207-W4111-Spring-2026-002-HW1b.ipynb

Introduction

Homework Overview

There are three parts to the homework.

- Part 1 (HW1A) walks you through the setup of your personal computer that is necessary for this course. You demonstrate completion by inserting screenshots or running code cells that demonstrate you successfully completed setup. You separately completed this homework and will submit on CourseWorks.

- Part 2 is a set of written questions that demonstrate you studied and understand lecture 1 and lecture 2 material from the course, and from the lecture 1 and lecture 2 slides from the recommended textbook.

- Part 3 is a set of practical questions demonstrating basic knowledge of relational algebra and SQL.

There are no track specific questions for homework 1. This notebook is part 2 and part 3.

Submission Instructions

The HW is due on Sunday, 08-February at 11:59 PM. Please follow the submission instructions on CourseWorks for submitting your homework.

Environment Setup

Execute the cells below. Make sure to set the password you used when installing MySQL in the cell below.

In [1]:
```
%load_ext sql
```

Set the password you used when installing MySQL below.

In [4]:
```
mysql_root_user = 'root'
mysql_root_password = ''
```

```
mysql_url = f"mysql+pymysql://{mysql_root_user}:{mysql_root_password}@localh
```

In [5]:
```
%sql $mysql_url
```

In [7]:
```
%sql use db_book;
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
```

Out[7]: []

In [8]:
```
%config SqlMagic.style = '_DEPRECATED_DEFAULT'
```

In [9]:
```
%sql select * from student where dept_name='Comp. Sci.';
```

```
 * mysql+pymysql://root:***@localhost
4 rows affected.
```

Out[9]:

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
54321	Williams	Comp. Sci.	54
76543	Brown	Comp. Sci.	58

Lecture 1 and 2 and Book Slides from Lecture 1 and 2 Questions

Q1

Question

Consider the applications used for managing courses and enrollments at Columbia, e.g. CourseWorks, Vergil. These applications use database management systems. Consider an alternate approach in which the university used a set of shared Google sheets to manage classes and enrollments. Faculty, administrators and students would edit the sheets to create classes, enroll in classes, etc. Identify five problems with the shared sheet approach and explain the features of a DBMS that solve the problem. Be succinct.

Answer

Problem 1: Data can be accessed by many users at once, which could result in unexpected interactions among users.
DBMS Solution: Maintains the isolation property by controlling concurrent access and preventing unexpected interactions.

Problem 2: Inefficient access due to a large number of concurrent users on the same instance of the DB.
DBMS Solution: Allows efficient access by controlling concurrent access and maintaining isolation.

Problem 3: Inability to store large amounts of data due to size limitations.
DBMS Solution: Store data remotely in efficient storage structures and only present data that has been queried.

Problem 4: Querying data is difficult and limited.
DBMS Solution: Provide a specialized querying language that exceeds capability of sheet formulas.

Problem 5: No specified schema, allowing maligned users to tamper with the data.
DBMS Solution: Strict database schema built into the DB definition forces users to interact with it correctly.

Q2

Question

What is data abstraction? What are the levels of data abstraction?

Answer

Data abstraction is the practice of hiding the implementation details of the DBMS and unnecessary information about the data from the user/stakeholder, presenting them only with the vital information present in the data. The levels of data abstraction are 1) the view level, 2) the logical level, and 3) the physical level.

Q3

Question

Briefly explain the concepts of database *schema* and *instance*.

Answer

The database schema is the logical structure of the database, i.e., the attribute set of an entity, the relationships between entities, etc. The database instance is a representation of the database at a given time. If the database were structured as a table, with entities represented by rows and attributes by columns, the database schema would correspond to the columns, while the database instance would correspond to the rows as a set.

Q4

Question

Consider a scenario in which multiple users and programs access a database. For this scenario, what is an example of the benefits of *physical data independence?*

Answer

Physical data independence ensures that the physical structure (e.g., organization of the data on the disk) of the data is independent of the logical schema. This means that if the physical structure changes for any reason (e.g., the filesystem is altered/reorganized in some way), the logical structure of the DB will stay constant. An example of this benefit would be when two users interact with the database, with the first migrating the DB to a different storage device, and the second being able to expect that they can interact with the DB in the same way as before, despite it existing on a different device.

Q5

Question

What are the two types/classes of languages that a DBMS system provides.

Answer

The two types of languages that a DBMS provides are 1) a Data Definition Language (DDL) and 2) a Data Manipulation Language. 1) is responsible for defining the database schema, while 2) is responsible for accessing and updating data in the DB.

Q6

Question

Briefly explain the concepts of *declarative* and *procedural* database manipulation languages. What is one benefit of declarative languages?

Answer

A procedural manipulation language requires the user to specify both what data are needed and *how* to get those data (e.g., through an algorithm). A declarative manipulation language requires the user to only specify what data are needed. A benefit of the latter is that they are easier to learn.

Q7

Question

What are the 3 levels of data modeling/entity relationship modeling?

Answer

The three levels of ER modeling are 1) the conceptual level, 2) the logical level, and 3) the physical level.

Q8

Question

What is a feature of the DBMS *storage manager* that provides fast access to data in a database?

Answer

A feature of the storage manager that provides fast access to data is the index data structure, which provides pointers to data items, allowing for quick lookups without needing to read entire tables.

Q9

Question

Briefly explain/define the concept of a transaction.

Answer

A transaction is a sequence of operations that together perform a single function in the database. The property of atomicity requires that all operations succeed for the function to complete (no partial completions allowed in DB). As an example consider student enrollment in a course. The sequence of operations that precede the enrollment include 1) checking student prerequisites, 2) checking seat availability, 3) adding student to

section, ..., n) charging tuition. For the student to be successfully enrolled, each operation must complete successfully. In this example, student enrollment represents the transaction in the database.

Q10

Question

Consider Vergil. Is this a *two tier* or *three tier* database application architecture?

Answer

Vergil is a three-tier database application architecture. This is because the user (e.g., the student, instructor, etc.) does not perform any direct database calls, but rather interacts with a GUI, which in turn communicates with a database system to access data.

Q11

Question

In the setup tasks from the previous section, you created a database schema and loaded data. You then wrote a simple query. Which tasks would a database administrator perform and which tasks would a database user perform?

Answer

A database administrator would perform the tasks of creating the database schema and loading the initial data. The database user, on the other hand, would perform the tasks of querying the data.

Q12

Question

Briefly explain the concepts of *unstructured*, *semi-structured* and *structured* data. Which type of data is a file holding an audio recording.

Answer

Structured data is data that has some sort of defined data model, format, or structure. An example of structured data is a database. Semi-stuctured data is one tier below structured data and typically has some structure to it but lacks a rigidly defined schema.

An example would be spreadsheets. Unstructured data has no inherent structure. An example would be different types of media, such as images and videos. A file holding an audio recording would be classified as unstructured data.

Q13

Question

For a file holding an audio recording, what would typical metadata be?

Answer

For a file holding an audio recording, typical metadata would include the file format (mp3, wav, etc.), the duration of the recording, and the file size, among other things.

Q14

Question

What are "The 5 Vs" of data?

Answer

The 5 V's of data are 1) Volume - the amount of data, 2) Variety - the diversity of data, 3) Velocity - the speed of data generation, 4) Veracity - the accuracy of the data, and 5) Value - the worth of data. They are used to characterize datasets and to inform decisions regarding database design.

Q15

Question

Briefly explain the concepts of *entity set, relationship set* and *attributes*.

Answer

An entity is an object distinguishable from other objects and represented by a set of attributes (e.g., a person with attributes first name, last name, etc.). An attribute is an indivisible (can't be further reduced) property of an entity (e.g., that person's last name). An entity set is a set of entities of the same type that share the same attributes (e.g., a set of people). Finally, a relationship set is a set of associations of the same type

between entities (e.g., "... is married to ..." could be a relationship set pulled from the entity sets of people).

Q16

Question

Explain the relationship between the $relational\ algebra$ and the SQL language.

Answer

SQL is a coding language that implements and extends the mathematical framework of relational algebra, allowing for more complex expressions and data manipulations that would otherwise be difficult or impossible to implement using relational algebra alone.

Q17

Question

The formal definition of a *theta join* is $r \bowtie_\theta s = \sigma_\theta(r \times s)$. Briefly explain how this definition relies on the $relational\ algebra$ being an $algebra$ and $closed$ under relational operations.

Answer

A core property of an algebra is that performing operations on the operands in the set will produce another operand in the same set, which is known as closure. In relational algebra, we perform operations on relations with the result that another relation is produced. Due to this property, we can compose functions and form another expression that obeys the rules of relational algebra. In the case of the theta join, it is defined as being equivalent to the composition of a Cartesian product with a select. The property of closure guarantees that this composition will yield another relation and thus be a legal expression.

Q18

Question

Briefly explain *natural join* and *theta join*. What is one "danger" associated with using a natural join?

Answer

The natural join is an operation that joins tables automatically based on a shared column. A theta join requires that the joining condition be specified. A "danger" associated with using a natural join is that columns that have the same name but that aren't necessarily related will be joined.

Q19

Question

Is it possible to express all valid SQL statements in the $relational\ algebra$? If not, give an example of an SQL statement that cannot be expressed in $relational\ algebra$.

Answer

No, it is not possible to express all valid SQL statements in relational algebra. There is no analog in relational algebra for the SQL statements INSERT, UPDATE, and DELETE.

Q20

Question

What is the result of executing the SQL statement `SELECT NULL=NULL`?

Answer

NULL

Practical Questions

P1

Question

Consider the following description of a datamodel.

There are three entity sets with the following attributes:

1. `Course`
   - `course_no`, which uniquely identifies a course.
   - `course_title`
   - `course_description`
2. Section

- `call_no,` which uniquely identifies a section.
- `course_no`
- `section_no`
- `semester`
- `year`
- `maximum_enrollment`

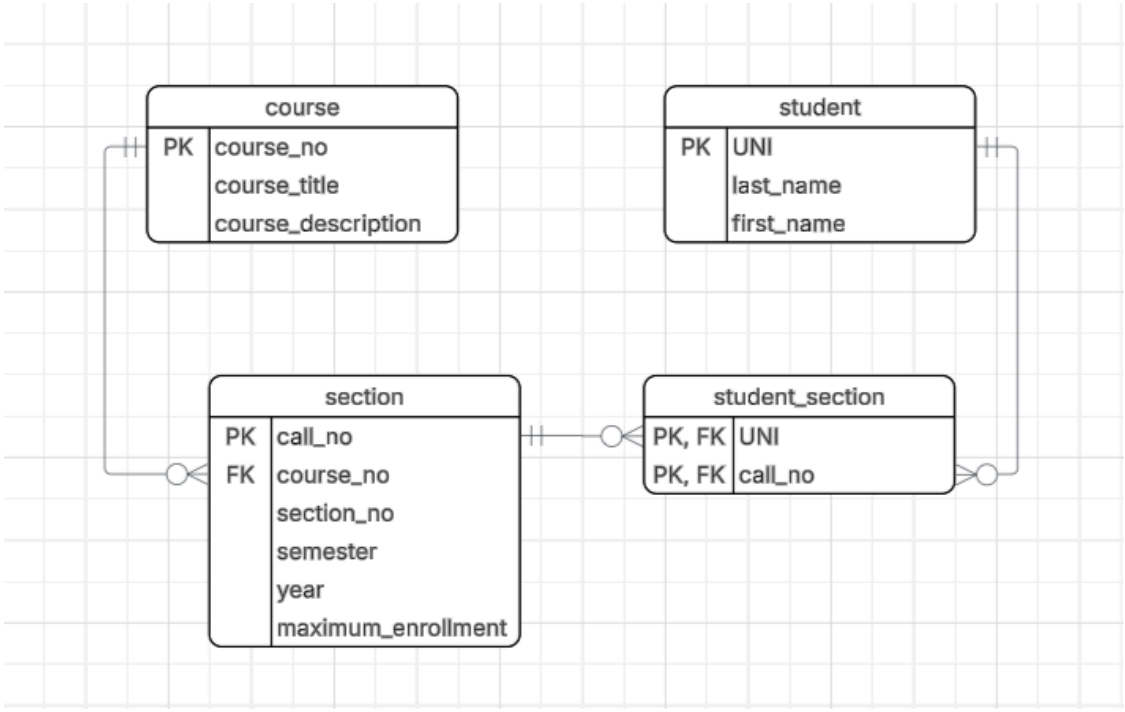
3. Student
   A. `UNI` , which uniquely identifies a student.
   B. `last_name`
   C. `first name`

A section is a section of exactly one course. A course may have 0, 1 or many sections. A student may be associated with 0, 1 or many sections. A section may be associated with 0, 1 or many students.

Using Crow's Foot Notation and Lucidchart or an equivalent tool, draw a logical datamodel for the scenario. Replace the image below with a screen capture of your diagram. Your data model must be realizable/implementable in SQL.

NOTE: The diagram below is a conceptual model. The question is asking you to define a logical model.

Answer



Logical Model

P2

Question

Using the RelaX relational calculator and the sample data that comes with the recommended textbook, write an algebra expression that computes the the `courses` that are in the "Comp. Sci." `department` and have `4` credits. Your answer should inly contain `course_id`, `title` and `credits.`

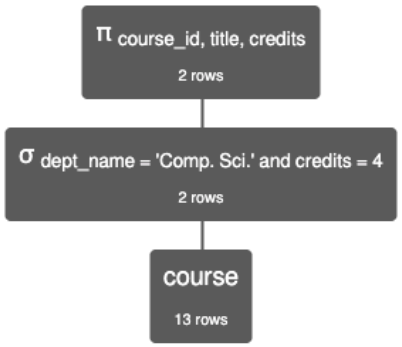
Your answer should be in the format below:

1. The text of your query.
2. A screen capture of the execution result.

The answer area below contains an example of the result format but for a different query.

Answer

```
π course_id, title, credits (σ dept_name = 'Comp. Sci.' ∧ credits = 4 (course))
```



π course_id, title, credits (σ dept_name = 'Comp. Sci.' and credits = 4 (course))

Execution time: 2 ms

course.course_id	course.title	course.credits
'CS-101'	'Intro. to Computer Science'	4
'CS-190'	'Game Design'	4

P2 Answer

P3

Question

Use the RelaX relational calculator and the sample data that comes with the recommended textbook for this question.

The following table lists instructors and the students they advise for instructors in the 'Comp. Sci.' department. Write a relational algebra expression that produces the table.

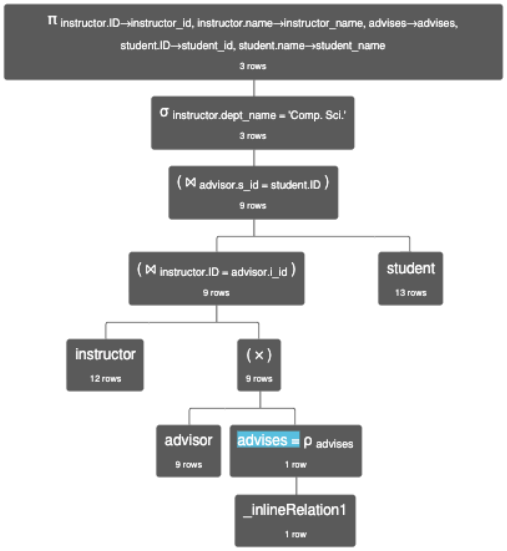
instructor_id	instructor_name	advises	student_id	student_name
10101	'Srinivasan'	'advises'	12345	'Shankar'
45565	'Katz'	'advises'	128	'Zhang'
45565	'Katz'	'advises'	76543	'Brown'

Your column names must match the above column names.

Replace the relational expression and image below with the result of executing your statement

Answer

```
π instructor.ID→instructor_id,
instructor.name→instructor_name,
advises→advises, student.ID→student_id,
student.name→student_name (
        σ instructor.dept_name = 'Comp. Sci.' (
                (instructor ⋈ instructor.ID = advisor.i_id (
                        advisor × ρ advises ({advises:string
                                'advises'
                        })
                )) ⋈ advisor.s_id = student.ID student
        )
)
```



π instructor.ID→instructor_id, instructor.name→instructor_name, advises→advises, student.ID→student_id, student.name→student_name (σ instructor.dept_name = 'Comp. Sci.' ((instructor ⋈ instructor.ID = advisor.i_id (advisor × ρ advises (_inlineRelation1))) ⋈ advisor.s_id = student.ID student))

Execution time: 4 ms

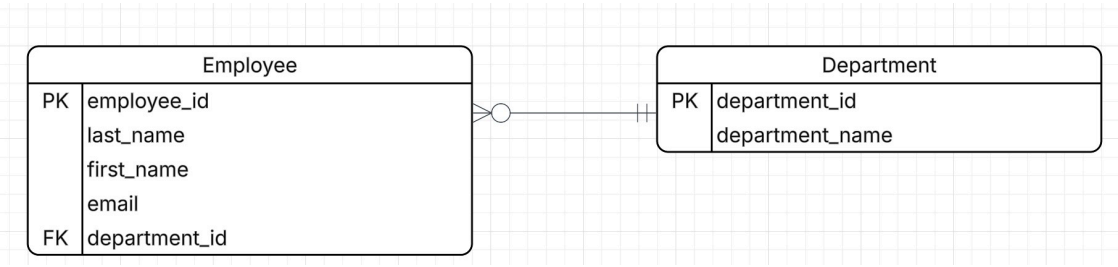
instructor_id	instructor_name	advises	student_id	student_name
10101	'Srinivasan'	'advises'	12345	'Shankar'
45565	'Katz'	'advises'	128	'Zhang'
45565	'Katz'	'advises'	76543	'Brown'

P2 Answer

P4

Question

The following is a simple Crow's Foot Diagram of two entity sets and a relationship. Write and execute `SQL CREATE TABLE` statements that implement the model in the diagram. You can assume that the data type for all attributes is `VARCHAR(64).`



P4 Diagram

Answer

In [10]:
```
%%sql

/*
    Put your create table statement below and execute this cell.
*/

CREATE DATABASE IF NOT EXISTS P4;

USE P4;

CREATE TABLE IF NOT EXISTS department (
    department_id varchar(64),
    department_name varchar(64),
    primary key (department_id)
);

CREATE TABLE IF NOT EXISTS employee (
    employee_id   varchar(64),
    department_id varchar(64) not null,
    last_name     varchar(64),
    first_name    varchar(64),
    email         varchar(64),
    primary key (employee_id),
    foreign key (department_id) references department
);
```

```
 * mysql+pymysql://root:***@localhost
1 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[10]: []

P5

Question

For the sample database you setup and loaded, write a `SQL SELECT` statement that returns the the courses that are in the "Comp. Sci." department and have 4 credits.

Answer

In [11]:
```
%%sql

/*
    Write your SQL below and execute the cell.
*/

USE DB_BOOK;

SELECT * FROM course
```

```
    WHERE dept_name = 'Comp. Sci.' AND
          credits = 4;
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
2 rows affected.
```

Out[11]:

course_id	title	dept_name	credits
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4

P6

Question

For the sample database, write and execute a statement that adds the following professor to the instructor's table.

```
{
    ID: 666,
    name: "Ferguson",
    dept_name: "Comp. Sci.",
    salary: 100000.00
}
```

Answer

In [12]:

```
%%sql

/*
    Write your SQL statement below and execute.
*/

INSERT INTO instructor
       VALUES(666, 'Ferguson', 'Comp. Sci.', 100000.0);

SELECT * FROM instructor
    WHERE dept_name = 'Comp. Sci.';
```

```
 * mysql+pymysql://root:***@localhost
1 rows affected.
4 rows affected.
```

Out[12]:

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
45565	Katz	Comp. Sci.	75000.00
666	Ferguson	Comp. Sci.	100000.00
83821	Brandt	Comp. Sci.	92000.00

P7

Question

For the sample database and the row you just inserted, write and execute an `SQL UPDATE` statement that is guaranteed to update ONLY the row you just inserted and set the salary to `120,000.00`.

Answer

In [13]:
```
%%sql

/*
    Write your SQL statement below and execute.
*/

UPDATE instructor
        SET salary = 120000.00
        WHERE ID = 666;

SELECT * FROM instructor
    WHERE dept_name = 'Comp. Sci.';
```

```
 * mysql+pymysql://root:***@localhost
1 rows affected.
4 rows affected.
```

Out[13]:

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
45565	Katz	Comp. Sci.	75000.00
666	Ferguson	Comp. Sci.	120000.00
83821	Brandt	Comp. Sci.	92000.00

P8

Question

For the sample database and the row you just inserted, write and execute an `SQL DELETE` statement that is guaranteed to delete ONLY the row you just inserted.

Answer

In [14]:
```
%%sql

/*
    Write your SQL statement below and execute.
*/
```

```
DELETE FROM instructor
        WHERE ID = 666;

SELECT * FROM instructor
    WHERE dept_name = 'Comp. Sci.';
```

```
 * mysql+pymysql://root:***@localhost
1 rows affected.
3 rows affected.
```

Out[14]:

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
45565	Katz	Comp. Sci.	75000.00
83821	Brandt	Comp. Sci.	92000.00

P9

Question

In this question, you use the `db_book` tables `student, instructor` and `advisor`. You are recreating the $relational\ algebra$ query from question `P3` in SQL. Enter your SQL in the cell below. My answer, which you must match, is below. The order of the rows does not matter.

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
3 rows affected.
```

[14]:

instructor_id	instructor_name	advises	student_id	student_name
10101	Srinivasan	advises	12345	Shankar
45565	Katz	advises	00128	Zhang
45565	Katz	advises	76543	Brown

Answer

In [15]:

```
%%sql

SELECT instructor.ID as instructor_id, instructor.name as instructor_name, '
    FROM (instructor JOIN advisor ON instructor.ID = advisor.i_id) JOIN stuc
    WHERE instructor.dept_name = 'Comp. Sci.';
```

```
 * mysql+pymysql://root:***@localhost
3 rows affected.
```

Out[15]:

instructor_id	instructor_name	advises	student_id	student_name
10101	Srinivasan	advises	12345	Shankar
45565	Katz	advises	00128	Zhang
45565	Katz	advises	76543	Brown

Create PDF

1. Use the `File -> Save and Export Notebook as -> HTML` option to save your notebook as an HTML file.
2. Copy the created HTML file to the same directory as the notebook and images you created.
3. Open the HTML file in a browser.
4. Using the *browser's* `File -> Print` option print the HTML file to a PDF. This is your submission format.
5. Follow the submission instructions on Ed Discussion to submit your file to GradeScope.

In [ ]:
```
!jupyter nbconvert *.ipynb --to html --embed-images
```