

## Exam — Databases

### Tasks A — Film Festival Management System

40 P.

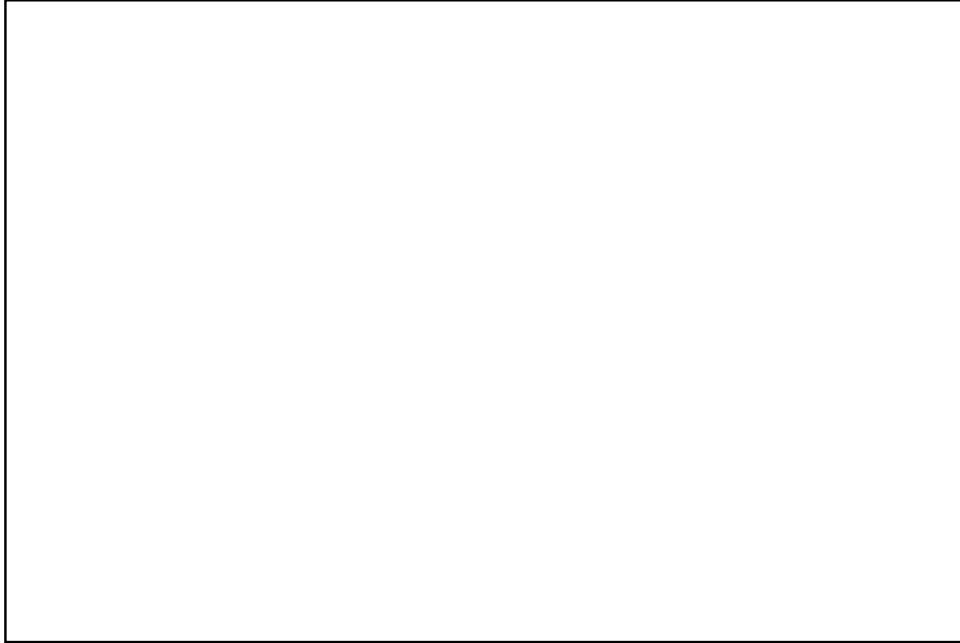
- **Context:** Design a system to manage an international film festival with multiple venues, screenings, films, directors, actors, and ticket sales.
- **Requirements:**
  1. Track films with attributes including title, release year, country, language, genre, and runtime
  2. Associate films with directors, producers, and cast members
  3. Manage multiple festival venues with different seating capacities
  4. Schedule film screenings across venues with no conflicts
  5. Handle ticket reservations and sales with different pricing tiers
  6. Track festival attendees and their ticket purchases
  7. Generate reports on attendance, popularity, and revenue
- **Deliverables:** Complete ERD with all entities, relationships, cardinalities, and a normalized relational schema

## Tasks B — Normalization

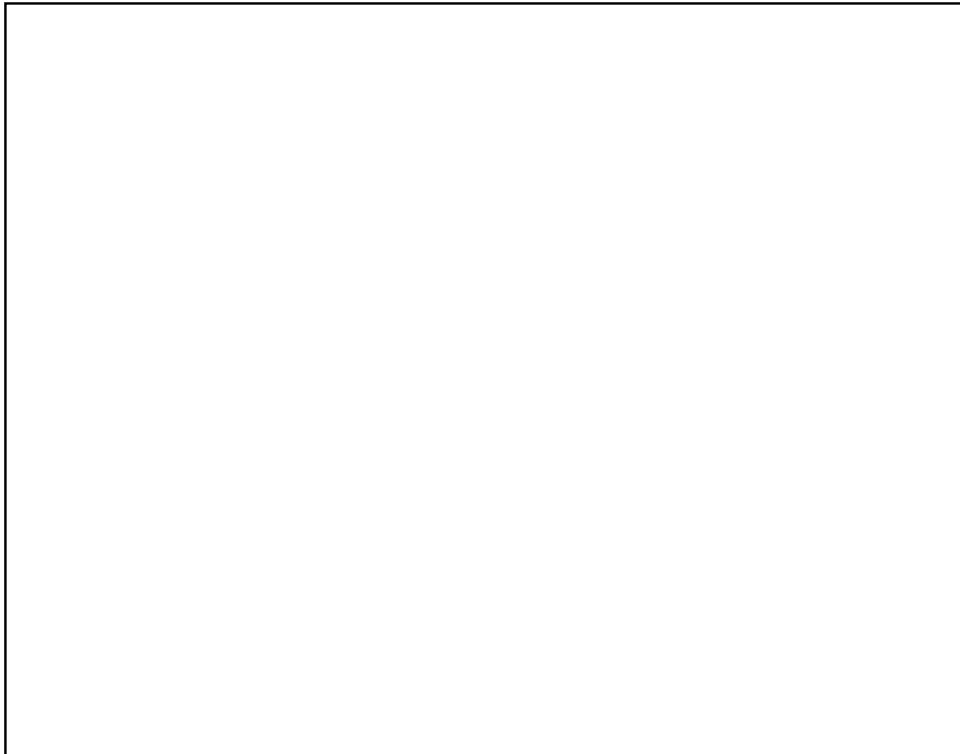
Consider the following schema:

```
1  CREATE TABLE bookstore (SQL
2      order_id INT,
3      order_date DATE,
4      customer_id INT,
5      customer_name VARCHAR(100),
6      customer_email VARCHAR(100),
7      customer_address TEXT,
8      customer_phone VARCHAR(20),
9      book_id INT,
10     book_title VARCHAR(200),
11     book_author VARCHAR(100),
12     book_publisher VARCHAR(100),
13     book_isbn VARCHAR(20),
14     book_category VARCHAR(50),
15     book_price DECIMAL(10, 2),
16     quantity INT,
17     payment_method VARCHAR(50),
18     card_number VARCHAR(20),
19     card_expiry VARCHAR(7),
20     shipping_method VARCHAR(50),
21     shipping_cost DECIMAL(10, 2),
22     discount_code VARCHAR(20),
23     discount_percentage DECIMAL(5, 2),
24     PRIMARY KEY (order_id, book_id)
25 );
```

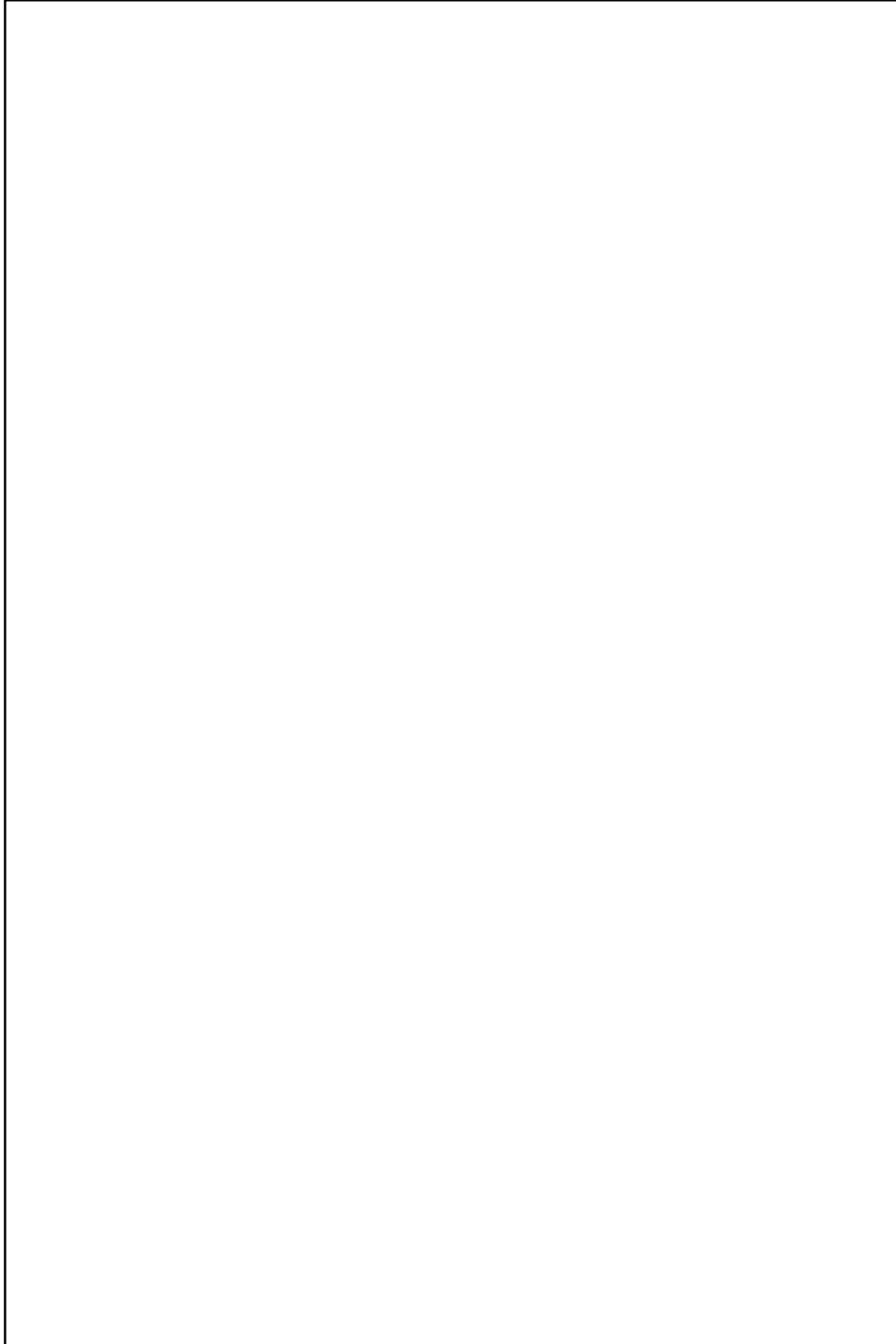
1. Describe what the type of data the schema contains and what the context of the schema most likely is. Give an example for both aspects. 4 P.



2. What is the problem with this schema? What kinds of issues does this type of schema have? How would you try to combat that? Give an example for an improvement! 4 P.



3. Improve the schema! Explain your decisions and implement at least 3 improvements for the schema in SQL. 22 P.



**Tasks C — General Terms of Databases**

10 P.

*Explain the following five terms that we encountered during the lecture in 2-3 sentences and give an example for each:*

- *Normalization*
- *Transaction*
- *ACID*
- *Selection*
- *Projection*

**Tasks D — True or False**

16 P.

For the following statements, decide whether each one is true or false and make a cross or checkmark in the respective column. For any correct answer, your point total will be increased by 2 points. For any wrong answer, your point total will be reduced by 2 points. You can receive negative points for this problem!

<b>Statement</b>	<b>True</b>	<b>False</b>
<i>The SQL keyword HAVING is used to filter records before they are grouped by the GROUP BY clause.</i>		
<i>Denormalization is always a poor design choice as it introduces data redundancy.</i>		
<i>The ACID properties (Atomicity, Collaboration, Iteration, Durability) are primarily relevant to transactions in databases because they describe how transactions are supposed to be.</i>		
<i>In an Entity-Relationship Diagram, a weak entity can exist independently of its identifying relationship.</i>		
<i>A relation is in Third Normal Form (3NF) if it is in Second Normal Form (2NF) and no non-prime attribute is transitively dependent on the primary key.</i>		
<i>A composite primary key can never contain a foreign key as one of its components.</i>		
<i>Database indexes always improve query performance regardless of the database size or query patterns.</i>		
<i>In PostgreSQL, a materialized view stores the query results physically and must be manually refreshed to update its data.</i>		

## Tasks E — University Management System Database Project

### Project Overview

Please design and implement a complete database system for university management, covering academic, administrative, and operational aspects. The project is divided into three sequential stages:

1. *Conceptual Design (ERD),*
2. *Logical Design (RM), and*
3. *Physical Implementation (SQL).*

#### 1. Stage: Entity-Relationship Diagram (ERD) Design

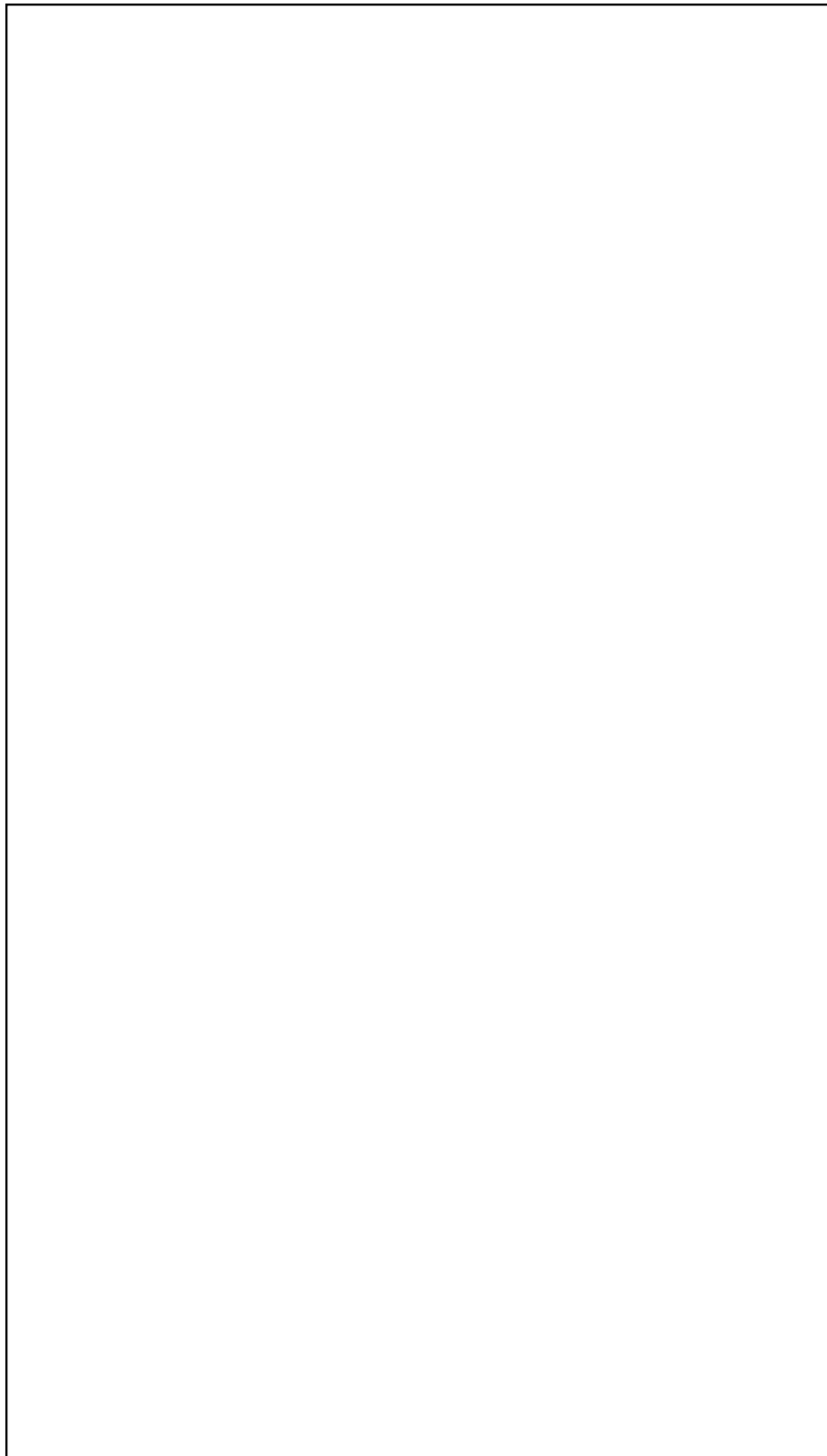
20 P.

##### Task Description

Design a comprehensive Entity-Relationship Diagram for a university management system. Your ERD must capture all significant entities, relationships, attributes, cardinalities, and constraints necessary for managing academic programs, student records, faculty, courses, registration, facilities, and departmental administration.

##### Requirements

- Identify all relevant entities with appropriate attributes
- Define meaningful relationships with proper cardinalities
- Specify primary keys and foreign keys
- Include derived attributes where appropriate
- Identify weak entities if necessary
- Document any assumptions made during the design process
- Use proper ERD notation (Chen, Crow's Foot, or UML)





## 2. Stage: Relational Model (RM) Design

20 P.

### Task Description

Transform your ERD into a complete Relational Model. For each entity in your ERD, define a corresponding relation with attributes, primary keys, foreign keys, and any additional constraints. Ensure your design is properly normalized (at least to 3NF) and document any denormalization decisions if applicable.

### Requirements

- Transform all entities and relationships into appropriate tables
- Properly handle many-to-many relationships with junction tables
- Define all primary keys, foreign keys, and constraints
- Specify attribute domains and data types
- Document functional dependencies
- Ensure the design is normalized to at least 3NF
- Explain any denormalization decisions with justification

### 3. Stage: SQL Implementation

20 P.

#### Task Description

Implement your relational model as a PostgreSQL database. Write the SQL statements to create all tables with appropriate constraints, and populate them with sample data. Then, write SQL queries to demonstrate the functionality of your database for common university management operations.

#### Requirements:

- Create DDL statements for all tables with proper constraints
- Include foreign key constraints with appropriate actions (CASCADE, SET NULL, etc.)
- Develop SQL queries for common operations like:
  - Course registration
  - Grade reporting
  - Faculty teaching assignments
  - Room scheduling
  - Student transcript generation
  - Department budget analysis