IMPORTANT: This exam is provided for study purposes only and may not be shared with students outside of the Winter 2024 CSCI2141 class.

# CSCI 2141: Intro to Database Systems

**Midterm Exam Section 1**

**Name:** _____

**Banner ID:** _____

Start: 1:10 PM

Latest arrival: 1:25 PM

Earliest departure: 1:30 PM

**End of exam: 2:25 PM**

| | |
|---|---|
| Part 1 | /20 |
| Part 2 | /20 |
| Part 3 | /20 |
| Part 4 | /20 |
| Part 5 | /20 |
| **Total** | **/100** |

**Midterm Rules:**

The midterm is comprised of five parts, each of which contains several questions. Each of the five parts is worth **20%** of the total.

The midterm is closed book. The following are offences that will result in your test being removed and a submission to the AIO:

- Use of any electronics (phones, smart watches, laptops, etc.)
- Use of any external references (textbooks and cheat sheets) are prohibited
- Talking during the exam
- Copying other students' answers or other forms of communication

**PART 1 – Multiple choice (8 questions; 2.5 points each)**

**Circle** the correct answer.

(1a) Which of the following statements concerning database models is **true**?

a) The internal model includes the memory addresses of database tables
b) The external model is only visible to the designer of a database
**c) The conceptual model shows relationships between entities**
d) The conceptual model does not include the names of entities

(1b) Which of the following is an inherent limitation of file-based data management?

a) The storage requirements are larger in file-based systems as each conceptual table is stored in a separate file
b) File-based data systems are more difficult to manipulate using simple Unix commands like 'grep' and 'sort'
c) File-based systems are more difficult to embed in complex workflows and server-side computation
**d) It is more difficult to ensure referential integrity amongst linked data sets**

(1c) Consider the following CREATE TABLE statement:

```
CREATE TABLE Zoo {
     zoo_id INT,
     zoo_name VARCHAR(100) UNIQUE NOT NULL,
     zoo_animal VARCHAR(100),
     zoo_animal_count INT,
     PRIMARY KEY(zoo_id),
     FOREIGN KEY (zoo_animal) REFERENCES Animal(animal_name),
     CHECK (zoo_animal_count >= 0)
}
```

Which of the statements below is **true**?

**a) zoo_animal can be NULL**
b) zoo_id can be NULL
c) zoo_animal_count can be a negative number
d) zoo_name can be NULL

(1d) What are essential properties of a foreign key?

a) Must uniquely identify each row in a table, can contain multiple candidate keys
**b) Must reference a primary key, can have the same value in multiple rows**
c) Must uniquely identify each row in a table, must be minimal, a single table can have more than one of these
d) Must uniquely identify each row in a table, must be minimal, a single table can have no more than one of these

(1e) Which of the following is a built-in function that performs aggregation?

a) HAVING
**b) SUM**
c) TIMEDIFF
d) CONCAT

(1f) Which of the following statements about joins is **true**?

a) INNER JOIN…USING can be applied to any columns in a pair of tables
b) A LEFT OUTER JOIN on columns $C_1$ and $C_2$ will always return at least one row that is returned by an INNER JOIN on $C_1$ and $C_2$
c) $T_1$ INNER JOIN $T_2$ …USING(*colname*) will always return columns in the same order as they appear in $T_1$ and $T_2$
**d) INNER JOIN…ON can be applied to different columns in the same table**

(1g) Which of the following is a disadvantage of using non-atomic attributes?

**a) Adding new values is likely to be complicated**
b) It is impossible to query whether a specific value is present.
c) Querying multivalued attributes is always slower than querying multiple tables with JOIN
d) MySQL does not allow comma-separated values for attributes

(1h) Which of the following actions may be necessary during a conversion of a table to first, second, or third normal form?

a) Creating a derived attribute
b) Constraining certain values to be NULL
**c) Creating a linker table**
d) Creating a transitive relationship

**PART 2 – Definitions / contrast (3 questions, 20 points total)**

2a (5 points): Describe how a LEFT OUTER JOIN works.

A LEFT OUTER JOIN returns all the rows from one table that match rows from the other table according to a matching criterion, plus any rows in the (first | left) table that do not match the second table.

2b (5 points): Define Candidate key.

A candidate key is a minimal set of attributes that can uniquely identify each row in a table.

2c (10 points): What is the difference between a *relation* and a *relationship*? What is another word for relation?

A relation is a single table.

A relationship describes the connections between tables, as established and enforced by foreign keys.

**PART 3 – Tables and Relationships (3 questions, 20 points total)**

Here are two tables.

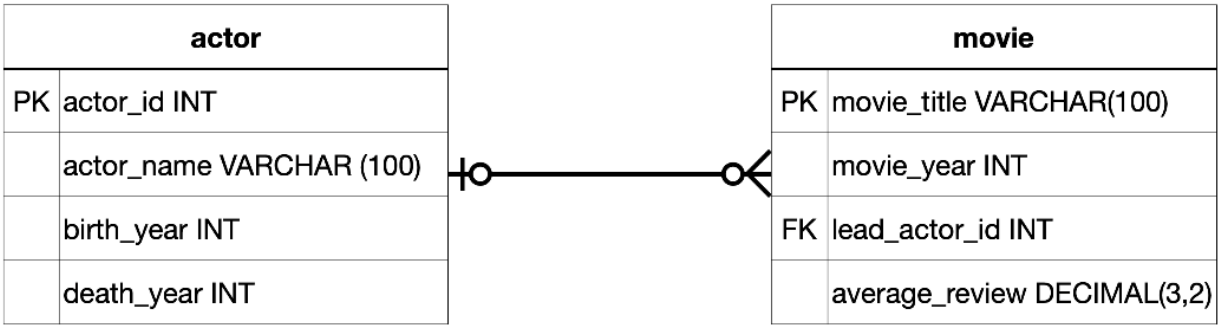
`actor`:

| actor_id | actor_name | birth_year | death_year |
|---|---|---|---|
| 1 | Marlon Brando | 1924 | 2004 |
| 2 | Meryl Streep | 1949 | NULL |
| 3 | Leonardo DiCaprio | 1974 | NULL |

`movie`:

| movie_title | movie_year | lead_actor_id | average_review |
|---|---|---|---|
| The Godfather | 1972 | 1 | 9.20 |
| Sophies Choice | 1982 | 2 | 7.60 |
| The Aviator | 2004 | 3 | 7.50 |
| Titanic | 1997 | 3 | 7.80 |

3a (10 points): Draw the internal model. Mark primary keys with “PK”, foreign keys with “FK”, and show relationships in Crow’s Foot notation.



3b (5 points): Write a CHECK constraint for one of the attributes in one of the tables. Explain what this constraint does.

CONSTRAINT `lifespan` CHECK ((death_year is null) OR (death_year > birth_year))

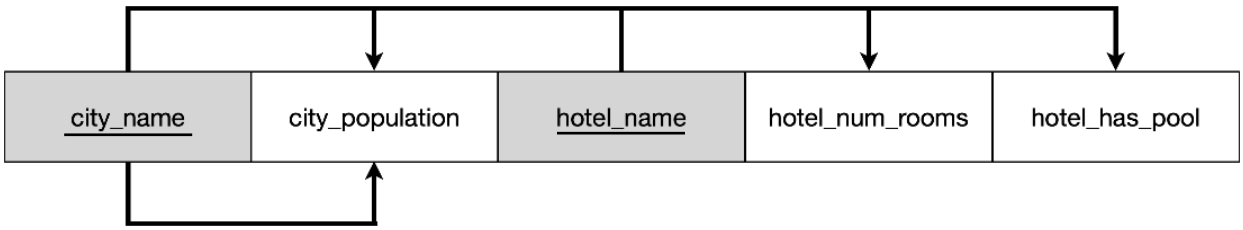
3c (5 points): What is the output of the following query:

```
SELECT movie_title, movie_year, actor_name, average_review
     FROM actor INNER JOIN movie
          ON (actor_id = lead_actor_id)
     WHERE average_review > 7.5;
```

| movie_title | movie_year | actor_name | average_review |
|---|---|---|---|
| The Godfather | 1972 | Marlon Brando | 9.20 |
| Sophies Choice | 1982 | Meryl Streep | 7.60 |
| Titanic | 1997 | Leonardo DiCaprio | 7.80 |

**PART 4: The normalization process (3 questions; 20 points total)**

Here is a dependency diagram showing a database table. Assume the table is in at least first normal form.




(4a) What are the two criteria we need to ensure (i.e., are not automatically enforced by the DBMS) that are necessary for 1NF?

1. Atomic values / no multi-valued attributes
2. No duplicate rows – enforced with primary key

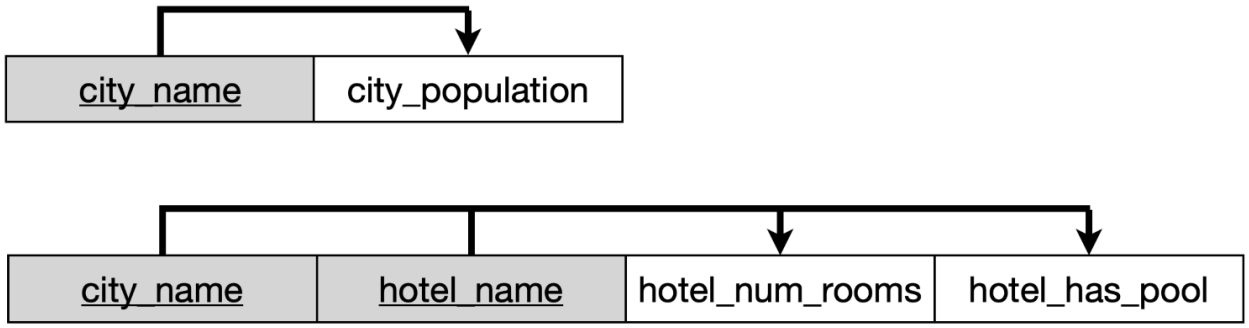
(4b) Is the table in 2NF? Why or why not?

The table is NOT in 2NF because there is a partial dependency (city_name -> city_population)

(4c) What is the key property of third normal form that distinguishes it from second normal form? Draw a dependency diagram that shows the structure of the table(s) once third normal form has been reached.

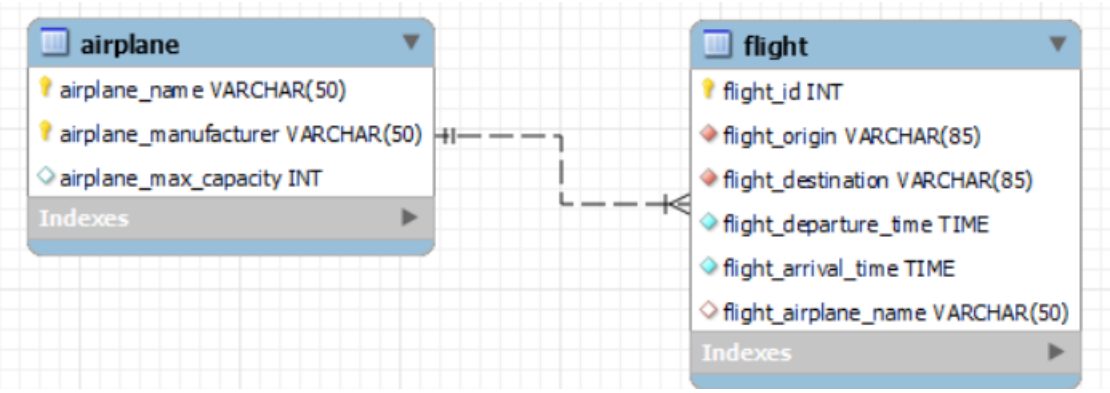
The key property of 3NF that differs from 2NF is elimination of transitive dependencies.

Dependency diagram in 3NF:



**PART 5 : SQL Queries (3 questions; 20 points total)**

Here is the schema for a subset of FlightDB:




5a (10 points): Identify the three errors in the following query and rewrite the query to fix them. Note that table aliases are <u>not</u> a problem in these queries.

```
SELECT airplane_name, airplane_manufacturer, COUNT(*)
      FROM airplane INNER JOIN flight
            USING (airplane_name)
      WHERE airplane_max_capacity > 200
      AND flight_origin >= "Halifax"
      GROUP BY airplane_name
      WHERE COUNT(*) > 3;
```

Error 1: airplane_manufacturer not in GROUP BY
Error 2: USING is not correct – this column is not present in both tables
Error 3: WHERE COUNT(*) > 3 should be HAVING

NOTE ABOUT SOLUTION: The solution below is one possibility. Other variations that corrected the above mistakes were accepted as well.

```
SELECT airplane_name, airplane_manufacturer, COUNT(*)
      FROM airplane INNER JOIN flight
            ON (airplane.airplane_name =  flight.flight_airplane_name)
      WHERE airplane_max_capacity > 200
      AND flight_origin >= "Halifax"
      GROUP BY airplane_name, airplane_manufacturer
      HAVING COUNT(*) > 3;
```

5b (2 points): Add a LIKE clause to the query to return only airplanes whose names start with “7”. Just write the LIKE clause, you don’t need to redo the entire query.

WHERE airplane_name LIKE “7%”

5c (8 points): Which of the following INSERT INTO statements will fail, and why? Assume all foreign-key constraints are satisfied, and there are no duplicate primary-key entries.

```
i) INSERT INTO flight
     VALUES (100,"Abu Dhabi","Dubai","08:00:00","10:00:00",NULL);

ii) INSERT INTO flight (flight_id, flight_origin, flight_destination,
     flight_departure_time, flight_arrival_time)
     VALUES (101,"Calgary","Dubai","08:00:00","10:00:00");

iii) INSERT INTO flight
     VALUES (100,"Abu Dhabi","Dubai","08:00:00","10:00:00");
```

`flight` has six attributes but only five are provided here. The correct INSERT INTO would need to specify which attributes are being defined.

```
INSERT INTO flight (flight_id, flight_origin, flight_destination,
flight_departure_time, flight_arrival_time)
     VALUES (100,"Abu Dhabi","Dubai","08:00:00","10:00:00");
```