Northeastern University
CS5200 – DBMS
Spring 2025, Derbinsky

Exam 1

Name: Erdun E

Problem	Points
General DBMS Knowledge	8 /10
The Relational Data Model	14 /20
SQL #1	2 /4
SQL #2	34 /36
SQL #3	10 /10
Security	3 /5
Bonus: Password Storage	0 /5
Total	71 /85

Instructions

- You will have 60 minutes to complete this exam; do **NOT** begin until instructed to do so.
- You are allowed to use one sheet of 8.5 × 11" paper for reference, as well as the provided SQL reference, but no other resources.
- No electronic devices may be used, including calculators, cell phones, cameras, and computers.
- Please write legibly: what I cannot read, I cannot award credit!

(10 pts.) General DBMS Knowledge

Respond to the questions below.

a) ~~DBMS~~ -2 is the declarative language used to define structure and manipulate data, as well as other objects (e.g., permissions), in a relational database.

b) For each description below, related to online purchases, enter the single best-matching `ACID` property (you must correctly write the full property name for credit):

Once an order is completed, the customer is able to post a review — Durability

A customer's order should never make product inventory negative — Consistency

Product inventory does not change if the payment method fails to authorize — Atomicity

Until it is purchased, many customers can hold a product in their carts — Isolation

(20 pts.) The Relational Data Model

Respond to the questions below.

a) Choose the single item from the right that best matches each item on the left.
Items on the right **may be used more than once** and may refer to Figure 1 below.

I	Row	A. Attribute
A	Column	B. Domain
A	Permissible atomic value(s) for a column	C. Foreign Key
E	Table	D. `NULL`
B	Table state: set/bag of ...	E. Relation
D	Entity Integrity: *invalid* value of primary key	F. Tuple
D	Unknown/not available	
C	Mechanism of referential integrity	
B	Explicit constraint: value must be within $[0, 4]$	
E	Figure 1: `Foo`	
A	Figure 1: `c`	
C	Figure 1: `arrow`	

-3

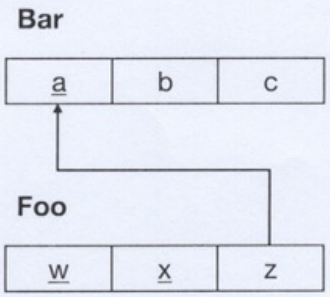


Figure 1: Relational Schema

b) Indicate the validity of each of the following statements by writing the complete word `true` or `false`.

False — A table's schema dictates how rows are ordered.

True — In a real database table without any keys, two rows can have the same values for all columns.

False — Because of the many features, a Relational Database Management System should *always* be used to manage an application's user data.

-1

c) List all potential primary keys for the current state of `Baz`: {z} ✓ missing! (express each response as a set {}).

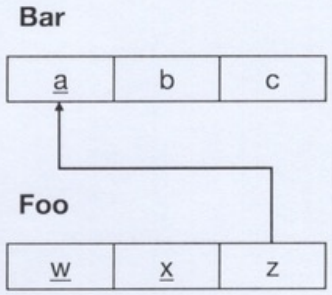
-2

Baz

w	x	y	z
1	a	α	vi
2	b	α	v
3	c	α	iv
4	a	β	iii
5	b	β	ii
NULL	c	β	i

(4 pts.) SQL #1

Consider the following relational schema reproduced from Figure 1.



Furthermore, assume...

- `Bar` has **10 rows**
- `Foo` has **50 rows** and `NULL` is not a permissible value of `z`

Characterize the result of the query...

```
SELECT * FROM Foo f INNER JOIN Bar b ON f.z=b.a
```

by indicating the number of...

6 columns

60 rows -1

and a brief description why...

6 columns = Foo has 3 cols, Bar has 3 cols ✓

60 rows = Foo has 50 rows, Bar has 10 rows ?. -1

(36 pts.) SQL #2

Consider the following database consisting of the `Users` *and* `AppRatings` *tables.*

Users

id	name	age
1	Alice	30
2	Bob	22
3	Cathy	50
4	Dylan	18

AppRatings

user	app	rating
1	Wordle	2
1	Spelling Bee	5
1	Connections	5
2	Wordle	3
2	Spelling Bee	5
3	Wordle	4

a) Find 4 errors in the DDL code to build the above database.

- Circle each error and label it with a number (1–4)
- In the corresponding line below, describe the problem

```
CREATE TABLE Users (
    id INT,
    name VARCHAR(10)
);

CREATE TABLE AppRatings (
    user INT PRIMARY KEY,
    app VARCHAR(20),
    rating INT PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Users (user)
);
```

1. didn't define the constraint to the age it's should be like age Int +2
2. app is primary key, but there don't add constraints for it
3. rating is not primary key. should delete it.

one issue +2

4. Wrong input. FOREIGN KEY (user) REFERENCES Users (id) +2

b) Draw the exact result produced from the following query:

```
SELECT
   ar.app AS a,
   AVG(ar.rating) AS b,
   MAX(u.age) AS c,
   COUNT(*) AS d
FROM
   Users u INNER JOIN AppRatings ar ON u.id=ar.user
GROUP BY
   ar.app
ORDER BY
   b DESC,
   d DESC,
   a ASC
```

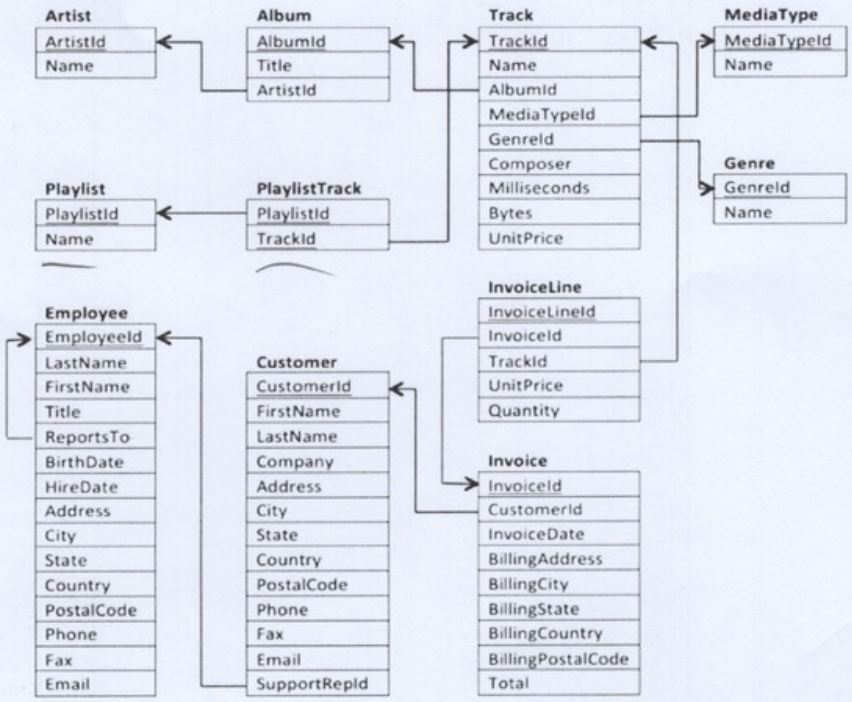
a	b	c	d
spelling Bee	5	30	2
Connections	5	30	1
Wordle	3	50	3

✓

−0

(10 pts.) SQL #3

Write an SQL query (valid in SQLite) against the Chinook database according to the prompt.



Multiple playlists contain the word "Classical" – determine the total number of tracks across all of them. (Note: a track can occur on multiple playlists and, if so, should be tallied multiple times.)

numClassical
150

```
Select
    Count(Track.TrackID) AS numClassical
From
    Track          -- not needed, but fine
    Inner Join PlaylistTrack on Track.TrackID = PlaylistTrack.TrackID
    Inner Join Playlist on PlaylistTrack.playlistID = Playlist.PlaylistID
Where
    Playlist.Name LIKE '%Classical%'
```

-0

(5 pts.) Security

Respond to the questions below.

Indicate the validity of each of the following statements by writing the complete word `true` or `false`.

True -1

Users are encouraged to select clever honeywords to prevent data breaches.

True -1

The `md5` hash function is considered effective for protecting sensitive information.

False ✓

It is safe to store passwords in plain text if a user-selected number ("salt") is also stored in plain text.

True

Data from past data breaches indicates that user-selected passwords are not random, which commonly makes dictionary attacks effective.

False

Manually quoting user-input values is the safest protection against SQL injection attacks.

(5 pts.) Bonus: Password Storage

Respond to the questions below.

Consider a user-login table that contains a password field. Now assume a well-intentioned database developer has just learned about secure password storage and so decides to append to each password, prior to hashing, a single randomly generated salt value (that is, one salt for the entire table, such as 42). Finally, assume an attacker gains access to this table of hashes. Answer the following questions related to password cracking in this scenario.

a) Explain the effect of this type of approach on preventing attack.

Hash Passwords, use salted hashes. e.g. bcrypt PBKDF2
Add Salting, Unique random values for each user to prvent precomputed attacts.

b) Now assume the attacker has successfully cracked a few passwords and inspects the results. How may the attacker more efficiently crack the remaining hashes.

Plaintext storage leads to vulnerabilites if compromised
Weak or common passwords are susceptible to dictionary attacks

c) Imagine you are newly hired as a database developer, and come across this salting technique. How do you improve this hashing policy? How will this impact current users of the system?

Hashing - One way transformation with no decryption (e.g. SHA-256, bcrypt)
Symmertric Encryption: One key for encryption and decryption
Asymmetric Encryption: Public key encrypts, private key decrypts
Encrypt sensitive data: (e.g. database files, backups, communications)