Final Exam: Written Questions

W4111 – Introduction to Databases: 20231COMS4111W002

# Guidance and Submission Instructions

# Foundation

## F1 – Benefits of DBMS

Before DBMS, organizations managed data by writing application programs that manipulated data in files. List 5 problems with the application-file approach and succinctly state of DBMS solve the problem.

Answer:

Problems with the application-file approach

1. **Data Redundancy** (data is usually stored across multiple files and has duplicates, leading to inconsistency or large storage size requirement):

DBMS solves the problem by centralising the data and allowing data to be shared between different files.

1. **Simultaneous Access Problem** (simultaneous access and manipulation to those files may cause data inconsistencies):

DBMS has concurrency control features that guarantee data integrity and consistency.

1. **Data Access Difficulty** (complex codes are required to retrieve and manipulate data in files, leading to data access difficulty).

DBMS offers a standardised query language (SQL) that simplifies data access and manipulation for all applications, making everything more accessible and consistent.

1. **Data Security** (those files can be accessed or manipulated easily, leading to data security concerns).

DBMS offers various security features, including access control, encryption and auditing, to protect the database and make sure that only authorised users can manipulate the data.

1. **Data Isolation** (data is stored in different files and/or formats. This makes it challenging to combine and analyse the data for users):

DBMS uses schema to well-define the data format, making it easier for users to integrate and/or retrieve data from various sources.

## F2 – Types of Data

Briefly explain structured data, semi-structured data and unstructured data. Give an example of each type of data.

## F3 – Physical Data Independence

What is physical data independence? What is a benefit?

## F4 – Concepts

Explain the following concepts and give an example of each:

1. Data manipulation language (DML)
2. Data definition language (DDL)
3. Procedure DDL
4. Declarative DDL

## F5 – Modeling

Briefly explain the following concepts and the role for each concept in data modeling. Give an example of the benefit of each level:

1. Conceptual model
2. Logical model
3. Physical model

## F6 – Application Architectures

Briefly explain:

1. Two-tier database application architecture
2. Three-tier application architecture

## F7 – Database Administrators (DBA)

List 5 tasks/functions that a DBA performs. Do DBAs typically use DDLs or DMLs?

# Relational Model

## R1 – Domain

Explain the importance of atomicity of domains. *float* is a type. An example of a *domain* might be a person’s *weight*. The type for *weight* might be a float, but give an example of how *float* is not the *domain.*

## R2 – Keys

Briefly define and explain the following concepts:

1. Superkey
2. Candidate key
3. Primary key
4. Foreign key

## R3 – Operators

The slides associated with the recommended text book list six basic relational operators

* select: 𝞼
* project: 𝝅
* union: U
* set difference: –
* Cartesian product: x
* rename: ⍴

Surprisingly, the list does not include *join: ⋈.* This is because join it is possible to derive join from a relational expression using more basic operators.

Briefly explain how to derive join from basic operators.

What is the importance of the relational algebra being *closed under the operators* for the derivation.

## R4 – Equivalent Queries

Briefly explain the concept of equivalent queries. Later lectures explained an important use of the concept. What is that use?

# SQL

## S1 – Foundation

Codd’s Rule 0 states

**Rule 0:** The *foundation rule*:

For any system that is advertised as, or claimed to be, a relational database management system, that system must be able to manage databases entirely through its relational capabilities.

Briefly explain and give examples of how the rule applies to:

1. Metadata
2. Security

## S2 – NULL

Codd’s Rule 3 states

**Rule 3:** *Systematic treatment of null values*:

Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.

Briefly explain the importance of the rule for:

1. Using different database schemas defined by multiple people.
2. SQL aggregation (group by) queries.

## S3 – Atomic Domains

The Columbia University directory of courses uses 20231COMS4111W002 for this section’s “key.” This is not atomic and is composed of:

* Year: 2023
* Semester: 1
* Department: COMS
* Course number: 4111
* Faculty code: W

Explain why having the non-atomic key creates problems for:

1. Integrity constraints.
2. Indexes

## S4 – JOINs

Briefly explain the following concepts:

* Natural join
* Equi-join
* Theta join
* Left join
* Right join
* Outer join

## S5 – Natural Join

Briefly explain how using the natural join might produce an incorrect answer.

## S6 – Views

List three benefits/use cases for defining views.

## S7 – Materialized View

What is a *materialized view?* List one advantage and disadvantage of a materialized view.

## S8 – View Updates

Explain two scenarios in view definition for which it is not possible to update the underlying tables?

## S9 – Primary/Unique

What is the main difference between a primary key constraint and a unique constraint?

## S10 – Cascade

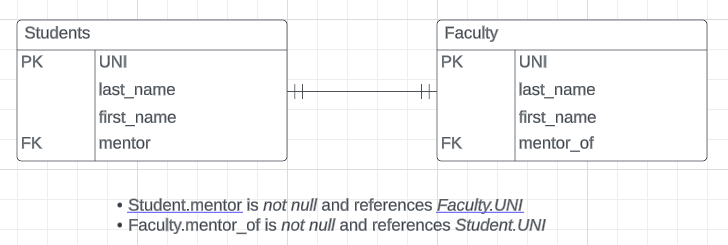
The *Classic Models* databases have several foreign key constraints. Two examples are:

1. *orders.customerNumber → customers.customerNumber*
2. *orderdetails.orderNumber → orders.orderNumber*

Briefly explain the concept of *cascading actions* relative to foreign keys. For which of the two examples above might cascading make sense?

## S11 – Foreign Keys and Transactions

Consider the logical data model below.



Some DBMS support deferring enforcing foreign key constraints until transaction commit. How would that capability help with the above model?

## S12 – Complex Check Constraints

Some databases do not support complex check constraints. Consider the following constraint:

check (time\_slot\_id in (select time\_slot\_id from time\_slot))

Assume the DBMS does not support subqueries in check constraints. What database capability would you use to implement equivalent functionality?

## S13 – Asset

What is the difference between an *Assert* constraint and a *Check* constraint?

## S14 – Types, Domains

Some relational DBMS support *user defined types* and *user defined domains.* Briefly explain the concepts and benefits.

## S15 – SQL Injection

Poorly written web applications can suffer from SQL Injection (Attacks). Briefly explain the concept.

## S16 – Functions, Procedures, Triggers

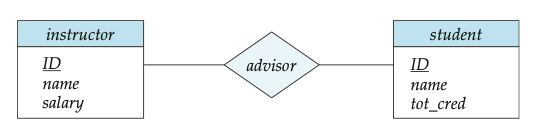
Briefly lists two differences between:

* Functions and Procedures
* Functions and Triggers
* Triggers and Procedures

# Entity – Relationship Modeling

## E1 – Implementing Relationships

The book’s entity-relationship modeling notation explicitly represents relationships. For example,



Crow’s Foot notation, which we used in class examples, does not support relationships. What type of entity did we use instead? Give two examples/reasons that require using the entity type.

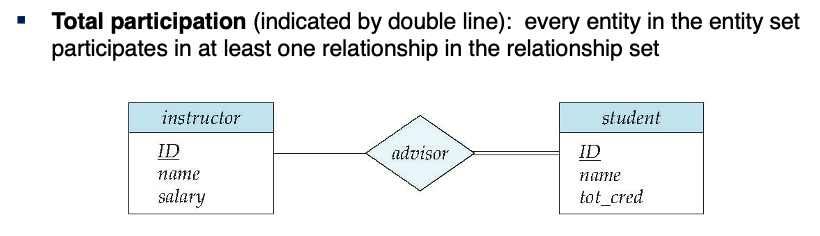
## E2 – Types of Relationships

Briefly explain the following concepts:

* Binary and Non-Binary Relationships
* Relationship Cardinality

## E3 – Participation

An important concept in ER modeling is *relationship participation.*



Use Lucidchart to draw an equivalent diagram in Crow’s Foot notation. What capability of SQL database definition would you use to enforce total participation?

## E4 – Weak Entity

Briefly explain the concept of a *weak entity.* Give an example from the Classic Models database.

## E5 – Specialization

Briefly explain the following concepts relative to implementing inheritance/specialization in an SQL schema.

* incomplete/complete
* disjoint/overlapping

# Normalization

## N1 – Duplicate/Redundant Data

A primary reason for schema normalization is to eliminate duplicate/redundant data. What are two problems that redundant/duplicate data can cause?

## N2 – Decomposition

Briefly explain the concept of *lossless decomposition* in normalization.

## N3 – Functional Dependency

Briefly explain the following concepts:

1. Functional Dependency
2. Closure of Functional Dependencies

## N4 – BCNF

Consider the sample university database that comes with the recommended textbook.

Consider a hypothetical relation:

*in\_dep (ID, name, salary, dept\_name, building, budget )*

Why is the relation not in BCNF?

## N5 – Third Normal Form

Briefly explain the difference between BCNF and 3rd Normal Form.

## N6 – Armstrong’s Axioms

Briefly list Armstrong’s Axioms for Functional Dependencies.

# Big Data

## B1 – MapReduce

Briefly define the following concepts in MapReduce:

1. Map
2. Reduce
3. Shuffle

## B2 – Algebraic Operation

Modern big data processing systems introduce the concepts of:

1. Directed acyclic graphs.
2. Algebraic operations.

Briefly the concepts.

Prof. Ferguson suggested in lectures some hypothetical algebraic operators for IMDB namebasics. An excerpt of the data is below. Suggest a couple of hypothetical operators to transform the data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **nconst** | **name** | **dob** | **dod** | **primaryProfessions** | **knownFor** |
| nm0000001 | Fred Astaire | 1899 | 1987 | soundtrack,actor,miscellaneous | tt0053137,tt0050419,tt0045537,tt0072308 |
| nm0000002 | Lauren Bacall | 1924 | 2014 | actress,soundtrack | tt0117057,tt0071877,tt0038355,tt0037382 |
| nm0000003 | Brigitte Bardot | 1934 |  | actress,soundtrack,music\_department | tt0056404,tt0049189,tt0057345,tt0054452 |
| nm0000004 | John Belushi | 1949 | 1982 | actor,soundtrack,writer | tt0077975,tt0078723,tt0072562,tt0080455 |
| nm0000005 | Ingmar Bergman | 1918 | 2007 | writer,director,actor | tt0050976,tt0060827,tt0083922,tt0050986 |
| nm0000006 | Ingrid Bergman | 1915 | 1982 | actress,soundtrack,producer | tt0036855,tt0038787,tt0034583,tt0038109 |
| nm0000007 | Humphrey Bogart | 1899 | 1957 | actor,soundtrack,producer | tt0042593,tt0037382,tt0034583,tt0043265 |
| nm0000008 | Marlon Brando | 1924 | 2004 | actor,soundtrack,director | tt0070849,tt0078788,tt0068646,tt0047296 |
| nm0000009 | Richard Burton | 1925 | 1984 | actor,soundtrack,producer | tt0061184,tt0059749,tt0057877,tt0087803 |
| nm0000010 | James Cagney | 1899 | 1986 | actor,soundtrack,director | tt0029870,tt0042041,tt0035575,tt0031867 |

## B3 – Concepts

Briefly explain the following concepts:

* Data Warehouse
* Data Lake
* Extract-Transform-Load

# Database Management System Implementation

## D1 – Storage Types

Briefly explain and list some differences between:

* RAM
* Solid State Drives
* Hard Drives

## D2 – Addressing

Briefly explain the concepts of:

* Logical block addressing
* Cylinder-Head-Sector addressing

## D3 – Elevator Algorithm

What is the elevator algorithm for disk arm scheduling and what is its benefit?

## D4 – Fixed Length Records versus Variable Length Records

Briefly define and list benefits of fixed length records and variable length records.

## D5 – BLOBs

Most application scenarios no longer use database BLOBs. What technology do applications typically use in place of BLOBs?

## D6 – File Organization

Give scenarios where:

* Sequential record organization is better than heap file organization.
* Multi-table clustering is better than sequential record organization.
* You would use table partitioning.

## D7 – Buffer Replacement Algorithm

For which type of query is most-recently-used a much better replacement algorithm than least-recently-used?

## D8 – Row Oriented versus Column Oriented

Explain why column oriented storage may be beneficial for scenarios in which:

1. Tables are large.
2. The only query operations are projection and aggregation.

## D8 – Index Types

Briefly explain the following concepts:

* Clustering index
* Dense index
* Sparse index

Can there be more than one clustering index on a table?

Must a sparse index be a clustering index?

## D9 – Hash versus B+ Tree

What is the primary benefit of a hash index relative to a B+ tree index? What are two disadvantages?

## D10 – Degree

Explain the relationship between key size, block size and B+ tree degree.

## D11 – Covering Index

What is a covering index and what is the benefit?

## D12 – Number of Indexes

What are two disadvantages of adding many indexes to a table?

## D13 – Buffering and Logging

Briefly explain:

* Force/No-Force policy
* Steal/No-steal policy
* The relationship between the policies and redo/undo logging.

## D14 – Access Path

Briefly explain the role of access path selection in query processing/optimization.

What is the “most selective index?”

## D15 – JOIN Optimization

Consider two tables L and R. Neither table is ordered and there are no indexes.

Consider the query *select \* from L join R using(c).*

If the tables were large, give a scenario for creating an index for optimization and the type of index.

What optimization might the query processor make if L was much, much smaller than R?

## D16 – JOIN Algorithms

Briefly explain the following concepts:

* Nested-loop join
* Block nested-loop join
* Indexed nested-loop join
* Merge-join
* Hash-join

## D17 – Optimization

Consider the following query on a very large table *people.*

*select last\_name, first\_name from people*

What single word/modification added to the query might motivate creating a has index for optimization?

## D18 – Optimization Techniques

Briefly explain the following concepts relative to query optimization:

* Operator selection
* Equivalent queries

## D19 – Equivalent Query Selection

Assume the tables in Classic Models were very, very large.

What is an equivalent query that a query optimizer might use in place of

SELECT

\*

FROM

customers JOIN orders USING(customerNumber)

WHERE

country = ‘France’ and status = ‘Shipped’;

## D20 – Locking

Briefly explain 2 Phase Locking and Strict 2 Phase Locking. What is the benefit of Strict 2 Phase Locking?

## D22 – Phantom

What is a “phantom” relative to database transactions/query processing?

## D25 – Serializable

Briefly explain serializability and conflict serializability.

## D26 – CAP

Briefly explain the CAP theorem.

## D27 – Consistency

Briefly explain *eventual consistency.*

## D28 – Sharing

Briefly explain database sharding and its benefits.

## D29 – Scaling

Briefly explain:

* Scale up versus scale out.
* Shared disk/data versus shared nothing/sharding.