Computer Science Databases:

A.1.1 Outline the difference between data and information

Data: Data is stored by computers, and can come in many forms such as integers, strings, Booleans. Data becomes information when it is interpreted and given context.

Information: Data that is given context to provide it with meaning.

A.1.2 Outline the differences between an information system and a database

Information Systems: A combination of information technologies and peoples activities using the technologies to support operations, management and decision making.

Database: They are a component of information systems

A.1.3 Discuss the need for databases

Databases can:

- Store a large amount of data: While spread sheets can store data, larger companies might have a large amount of data that a spreadsheet cannot contain, which is why a database is then important as it can store large amounts of data.

- Ease the updating of data: Multiple people cannot edit the same file at once, but multiple people can edit the same database at once.

- Improves accuracy: When a user is entering data, there is no process that validates this data. Which might cause issues where data is incorrectly spelt, the date is wrong, or the data itself is wrong. Databases solve this issue through data validation.

- Improve security: Spreadsheets and normal files do not have any security, allowing unauthorized users to easily get access for the data. Databases provide better security for the individuals and organizations that require them.

- Reduce redundancy: In a spreadsheet, it is possible to add the same data multiple times, causing redundancy. A database solves this issue through a primary key that ensures duplicates of data are not possible.

A.1.5 Define the term database transactions

Transaction: A logical unit of work executed to retrieve or update data. For a transaction to occur, it must exhibit the four properties of **ACID** (Atomicity, Consistency, Isolation, Durability)

A.1.4 Describe the use of transactions, states and updates to maintain data consistency and integrity

States: They are the states that a transaction goes through, they inform the DBA about the current state of the transaction whether it committed or was aborted.

How do they ensure data consistency and integrity?

Consistency in transactions refers to transactions never violating rules that dictate what data is valid and not valid in a database. Whenever a transaction breaks one of the rules and introduces data that is inconsistent, it will roll back.

Data integrity is the overall completeness, accuracy and consistency of data. It is ensured in the creation phase of the database, where certain rules are set in place to ensure no data or updates are repeated, and all data follow the same format.

A.1.6 Explain concurrency in data sharing situation

Concurrency is defined as operating two actions at the same time. In databases, two users can try to update the same data simultaneously. This might cause issues regarding data integrity, which is why databases place partitions and rules that do not allow concurrency when updating the same data.

A.1.7 Explain the importance of the ACID properties of a database transaction

Atomicity: The entire transaction takes place at once or it doesn't happen

Consistency: The database and its contents must be consistent before and after the transaction

Isolation: Multiple transactions can occur independently without interference

Durability: The changes of a successful transaction occur even if the system failure occurs

A.1.8 Describe the two functions databases require to be performed on them

Query Functions: A request to view data or information from a table or combination of tables. This request can be specific to a certain extent that the user is looking for such as a certain age, location, or any value in the database

Update Functions: Used to modify the existing records in a table

A.1.9 Explain the role of data validation and verification

Validation: To check the data being inputted to make sure it matches the rules and requirements set for current data in the database

Data Verification: To ensure that the user is inputting the data that they want to input, such as making the user input their password twice to make sure they didn't make a spelling mistake.

A.2.1 Define the terms Database Management System (DBMS) and Relational Database Management System (RDBMS)

Database Management System: A system software used to create and manage a database. It provides the user with a systematic way to create, retrieve, update and manage data

Relational Database Management System: It is a DBMS based on the relational model. The data in a RDBMS is stored in a table, consisting of fields (UserID, UserAge, UserName) and records where each record is an individual entry filling out the fields.

A.2.3 Explain how a DBMS can be used to promote data security

Access rights: Allowing access to the data through a PIN or password. The access can vary as it can be read-only or read and write.

Audit trials: A trials that keeps up with everyone that accessed the data and edited it

Data locking: When multiple transactions on the same data are occurring concurrently, data locking is requested where certain data cannot be edited.

Validation: Previously talked about, ensures that the data inputted follows the requirements and rules set by the database.

Encryption: An algorithm is used to encrypt the data where it all becomes unreadable code that can only be decrypted through a key also generated by the algorithm provided to the user.

Backups: When a file is corrupt, it is safer and easier to delete that file and use a backup of it.

A.2.2 Outline the functions and tools of a DBMS

- Data Dictionary Tool Management

- Data Definition Language

- Data storage management

- Data integrity management

A.2.6 Outline the nature of a Data Dictionary

A data dictionary tool is an inventory that holds all information about the fields of a RDBMS such as the type of data that should be inputted into it, a limit on the number of characters or numbers, description, and any other information that could be of use.

A.2.7 Explain the importance of a Data Definition Language (DDL) in implementing a data model

DDL is a programming language used to define and modify a database. DDL's can be used to create the database, insert the data into it then update its contents. It is important as it is one of the only methods for a programmer to communicate with a database to update and store data permanently.

A.2.8 Explain the importance of data modelling in the design of a database

A database model is a type of data modelling that defines the logical structure of a database. It determines the relationships between data and the constraints in order to gain a better idea on how to store, update and retrieve the data.

Data modelling is important as it results in:

- Reduced costs

- Higher quality

- Clearer scope

- Fewer errors

- Managed risk

A.2.4 Define the term schema

Schema: The organization of data as a blueprint of how the database is constructed. It includes table names, field names, data types and the relationship between them.

A.2.5 Identify the characteristics of the three levels of the schema: conceptual, logical, physical

Physical

- Physical storage structure of the database

- Low level data structures in detail

Logical

Conceptual:

- It describes the part of the database that the user is interested in and hides the remaining database from the user

A.2.9 Define the following database terms: table, record, field, primary key, secondary key, foreign key, candidate key, composite primary key, join

Table: A relation or a file

Record: A tuple or row

Field: An attribute or row

Primary key: A primary key is a keyword of a certain type that is unique to each record used to identify that record from the rest. Typically, the primary key makes up a column and is and ID.

Secondary key / Candidate key: It is a column that is eligible to be a primary key but has not been chosen, so is called a secondary key or a candidate key.

Foreign key: A foreign key is a column in a relational database that that provide a link between two tables by referencing a primary key in another table.

Composite primary key: A combination of fields that when both used together can identify each record uniquely. Commonly used when a singular column cannot identify each record uniquely, but a combination of them can.

Join: An SQL operation used to create a link between two tables based on matching columns. Used a foreign and primary key.

A.2.10 Identify the different types of relationships within a database: one-to-one, one-to-many, many-to-many

One-to-One: Each record in a table is associated with one and only record in another table.

One-to-Many: Each record in one table can be associated to many records in another table.

Many-to-Many: Multiple records in a table are associated with multiple records in another table. In this case, a join table is used where foreign keys are implemented from each table.

A.2.11 Outline the issues cause by redundant data

- Data integrity

- Data reliability

- Space

- Time

A.2.13 Describe the differences between 1st Normal Form (1NF), 2nd Normal Form (2NF) and 3rd Normal Form (3NF)

A.2.14 Describe the characteristics of a normalized database

A.2.17 Construct a relational database to 3NF using objects such as tables

What is normalization?

It is the reconstruction of data to get rid of any redundancy and all data dependencies are logical such that related data items are stored together.

First Normal Form (1NF):

- Each field must contain a single atomic value

- Every field must be uniquely named

- Each row is unique and has a primary key such that no records have repeating data

Second Normal Form (2NF):
- Table must be in 1NF

- Each non-key attribute must depend on the primary key

Third Normal Form (3NF):

- Table must be in 2NF

- Each non-key attribute cannot depend on another non-key attribute

What are the differences between the forms?

- 1NF has no repeating fields and records

- 2NF is based on full functional dependency

- 3NF is removes transitional dependency

A.2.12 Outline the importance of referential integrity in a normalized database

What is referential integrity?

It is the relationship between tables. The primary key of a table can appear in another tables as a foreign key.

What is the importance of referential integrity?

- Prevents the entry of duplicate data

- Prevents one table to point to a nonexistent field in another table

- Guarantees consistency between the tables

- Prevents the deletion of a record referred to by a foreign key

- Prevents the creation of a record in a table that has a foreign key unless there is a primary key