Databases (1.3.2)

A database: An organised collection of data.

Electronic databases make it:

- Easier to add, delete, modify and back up data
- Easier for it to be accessed by multiple people at the same time and from different locations.

Flat-file database: Contains only one table. Flat file databases are very simple, quick to set up, and require little expertise to maintain. A large flat-file database will contain lots of repetitive data (data redundancy, this is unnecessary). Data integrity decreases (if a person changes address, for example, data must be updated in several places). It is slow to query.

A relational database: Where there are multiple tables linked together with relationships. We need a common field in both tables for the relationship to work.

The primary key: a unique identifier. A field that uniquely identifies each record within a table. A Foreign key: where the primary key of one table is a field in another table, creating a relationship.

Relationships come in three forms, 1:1 (one-to-one), 1:M (one-to-many), M:M (many-to-many). M:M relationships make logical sense, but they are bad. Entity-relationship diagram (ERD): Used to represent database relationships.

TableName (<u>Primary Key Field</u>, Other Fields)

Paper-based forms

- Data input is manual
- A human reads the form and types the data into a computer

Optical Character Recognition (OCR)

- Reads text by interpreting the shapes of letters. Works far better with printed text than handwriting

Optical Mark Recognition (OMR)

- Very fast and efficient, significantly reduces human error

DBMS systems:

- Enforce validation rules
- Provide secure access
- Manage multiple users

JSON: A human-readable, common standard format for storing and transporting data. A CSV file is another popular format for exchanging data. Manual methods for transferring data include Email and a USB Memory stick.

Electronic Data Interchange (EDI): Where two databases interface with one another so they can read and write directly to and from each other's tables; a protocol between two systems to facilitate the exchange of data. EDI significantly increases the speed and efficiency of data transmission. There is no human involvement, however, any error in the data will be replicated across multiple systems.

Normal Form: A label for a method of arranging data in database tables. Normalisation: The concept of splitting tables in a database, arranging the data to move it from 1NF to 2NF to 3NF, and removing data redundancy and inconsistency.

First Normal Form (1NF): All field names must be unique, Values in fields should be from the same domain, Values in fields should be atomic, No two records can be identical, Each table needs a primary key
Second Normal Form (2NF): The data is already in 1NF, Remove partial dependencies, Remove M:M relationships
Third Normal Form (3NF): The data is already in 2NF, Remove transitive dependencies

Databases (1.3.2)

Transitive Dependency: "All fields are dependent on the key, the whole key and nothing but the key."

Dependency: A relation between two or more attributes. Partial dependency: where one or more of the fields depend on only part of the primary key. This can arise if the primary key is a composite key. If a field or group of fields can be inferred from a field that isn't the primary key, they should be split into a separate table.

Fix a M:M relationship:

- Create a linking table, with the two initial primary keys as the composite key
- Join the tables with 1:M relationships

Transaction: A single, logical, indivisible unit of work. Transaction processing: Any processing that is divided into transactions. Serialisation: When multiple transactions are executed concurrently, the effect should be the same as if they had been executed serially. Record Locking: The DBMS will lock affected records until updates are completed, causing slight delays to access the data. However, when several records must be locked, a deadlock can occur.

- The Timestamp Protocol:
- Each record has two timestamps: when it was last read, and when it was last updated.
- All operations are processed in timestamp order
- For a write operation:
    - If either record timestamp is newer than the timestamp of the operation, reject the operation
    - Otherwise, allow the operation, set both record timestamps to the operation timestamp
- For a read operation:
    - If the write record timestamp is newer than the operation timestamp, reject it
    - Otherwise, allow the operation, set the read record timestamp to the operation timestamp

Databases have user access rights.

Data Integrity: The process of maintaining the consistency of the database. Referential Integrity: The process of maintaining the consistency of data within a relationship. If you want to delete an object, it must be deleted from every table, to maintain data integrity. Manually, the procedure is prone to mistakes. Enforce a cascade delete restraint on relationships. If we delete an object from one table, all associated rows will also be deleted. However, this may permanently delete important records. Similarly, you cannot add records to a table without adding matching records in other tables.

These four rules ensure data integrity:

- Atomicity: A transaction is performed completely or not at all.
- Consistency: A transaction must retain the overall state of the database and not violate data integrity constraints.
- Isolation: The effect must be that each transaction is executed serially (through record locking)
- Durability: Transactions are permanent (write changes back to permanent secondary storage immediately, instead of storing in RAM)