Relational Database

definitions are highlighted!

Database

- A collection of data items and links between them, structured so it can be accessed by different applications
- Provides a system for storing, accessing and managing data
- Benefits of usage:
    - Data integrity
    - Data security
    - Efficient data retrieval
    - Support for complex queries and analysis
- Simplest form: Flat file
    - A single table stored in a text field, where each row is a record and each column is a field.
    - Advantages:
        - Easy to create and work with for small-scale applications
        - Application programmes can be written to access flat files using any programming language.
    - Disadvantages (for larger sets of data with more complex relationships):
        - Harder to maintain
        - Managing links between data and performing advanced queries can be inefficient
    - To counter these disadvantages, for larger sets of data, relational databases are used.

Flat-file database	Relational database
Data is stored in a single table or file.	Data is stored in multiple linked tables.
No built-in relationships between data sets; links must be created manually.	Relationships between tables are established using keys (primary and foreign).
Changes to the structure require manual updates; there is no automatic enforcement of data constraints.	Structure changes and constraints are handled automatically; updates in one place are reflected throughout the database.
Often requires custom programmes or scripts to retrieve specific information	Query languages (e.g. SQL) allows easy data retrieval and manipulation.
Higher risk of data redundancy and inconsistency	Normalisation reduces data redundancy and improves consistency.
Lesser scalability	Greater scalability
Lesser flexibility	Greater flexibility

Relational database

- Data is stored in relations and represented as tuples (rows).
- A collection of relational tables
    - **Table:** A 2-dimensional representation of data stored in rows and columns
    - Description can be expressed as:
        - TableName (Attribute 1, Attribute 2, …)
    - Made up of records and fields
        - Record — a complete set of data about a single item

        - Field — represents a specific attribute or property of the data
            - Holds a single atomic value within a record
            - Associated with a column in a table
    - A table is a relation if:
        - Its values are atomic.
        - Its columns are of the same kind.
        - Its rows are unique.
        - The order of columns is insignificant.
        - Each column has a unique name.
- Keys
    - Candidate key
        - *A minimal set of fields which can uniquely identify each record in a table*
        - Should never be NULL or empty
        - There can be more than one candidate key in a table, and it can also be a combination of multiple fields.
    - Primary key
        - *A candidate key that is the most appropriate to become the main key for a table*
        - Tells a particular record apart from others
        - Should not be changed over time
        - Does not allow duplicate or null values
    - Secondary key — *a candidate key that is not selected as the primary key*
        - Provides fast access when searching in that attribute
        - Cannot have duplicate or null values as it is a candidate key (which is supposed to be unique)
    - Composite key
        - *A combination of 2 or more fields in a table that can be used to uniquely identify each record in a table*

        - Uniqueness is only guaranteed when the fields are combined, not when they are taken individually
    - Foreign key
        - *A field in one table that refers to the primary key in another table*
            - Thus, it cannot contain values that do not appear in the table that it refers to.
        - Can be used to cross-reference tables, so cannot contain values that do not appear in the table that it refers to
        - Allows duplicate and null values
- Entity-Relationship (E-R) diagram
    - Entity — a specific object of interest
        - Usually named with collective nouns
        - Represented by rectangles
    - Relationship — a link between 2 entities
        - #1: One to One



        - #2: One to Many



        - #3: Many to Many



Data redundancy

- *A data organisation issue that allows the unnecessary duplication of data within the database*
    - Essentially, the same data being stored more than once.
- Insertion anomaly — occurs when data cannot be added to the database because other required data is missing
- Update anomaly — occurs when changes to duplicated data are only partially made, leading to inconsistencies
- Deletion anomaly — occurs when deleting a record unintentionally removes valuable data that should have been retained

Data dependency

- Functional dependency
    - If, for every valid instance of X, the value of X uniquely determines the value of Y, Y is functionally dependent on X.
    - Represented as X —> Y
- Transitive dependency
    - A dependency that is indirectly formed between 2 entities
    - If Y is functionally dependent on X but X is not functionally dependent on Y, and Z is functionally dependent on Y, Z is transitively dependent on X.

Normalisation

- The process of organising the tables in a database to reduce data redundancy and prevent inconsistent data
- Tables should be sorted in a way that:
    - No data is unnecessarily duplicated.
    - Data is consistent through the database (no update anomaly).

    - The structure of each table is flexible enough to allow you to enter as many/as few items as you want to (no insertion anomaly).
    - The structure enables a user to make all kinds of complex queries relating data from different tables.
- First normal form (1NF)
    - **Condition:** All columns must be atomic; information in each column cannot be further broken down.
- Second normal form (2NF)
    - Conditions:
        - The table should be in 1NF.
        - Every non-key attribute must be fully dependent on the entire primary key.
            - Must be considered when there is a composite key
    - **Note:** Whenever there is a many-to-many relationship between 2 entities, a link table is always needed in the middle to bring them to 2NF form.
- Third normal form (3NF)
    - Conditions:
        - The table should be in 2NF.
        - The table should not have transitive dependencies.
- In standard notation, the primary/composite keys are indicated using a solid underline, while foreign keys are indicated using a dashed underline.
    - e.g.

```
Student (MatricNo, Name, Gender, CivicsClass, CivicsTutor, HomeRoom)
StudentCCA (MatricNo, CCAName)
CCAInfo (CCAName, CCATeacherIC)
```

@star.notes