

# FACTFILE: GCE DIGITAL TECHNOLOGY

## UNIT A2 1: INFORMATION SYSTEMS



### Databases 1

#### Learning Outcomes

**Students should be able to:**

- describe the main features of a relational database.
- describe the difference between a logical data model and a physical model.
- define key terms attribute, entity, primary key, composite key, foreign key, relationships, referential integrity, data duplication and data inconsistency.
- produce an entity-relationship (ER) model from the given data requirements for a scenario.

#### Content in Databases

- The main features of a relational database.
- The differences between a logical data model and a physical data model.
- Definition of the key terms attribute, entity, primary key, composite key, foreign key, relationships, referential integrity, data duplication and data inconsistency.
- How to produce an entity-relationship (ER) Model

#### Databases

A database is a collection of data items structured so that data can be searched and retrieved when necessary. An address book containing names, addresses and telephone numbers is an example of a simple database.

Data can be held in a flat file structure where the entire database is stored in a single file and there is one row for each record and one column for each field. This type of structure leads to data being repeated unnecessarily (redundancy) and therefore it is not efficient. Relational databases store data more efficiently.

For example, the table below contains information about employees who are working on different projects. This structure could be considered a flat file.

Consider the table below.

Field							
Project No	Project Name	Employee No	Name	Start Date	Pay Grade	Hourly Rate	Hours Allocated
1200	Tanya Travels Website	10	Margaret Lynch	12/12/15	A	£90	25
1200	Tanya Travels Website	12	Gerry Mattwell	09/07/16	B	£80	20
1200	<b>Tanye</b> Travels Website	19	Ruth Debson	20/08/14	C	£70	30
1902	Action Pursuits Website	10	Margaret Lynch	30/03/16	A	£90	30
1902	Action Pursuits Website	18	Alex Kay	01/06/15	B	£70	25

Record

This table is made up of records. A single record is represented by a row in the table. Each record is made up of a set of fields. Each field is represented by a column in the table.

Notice that Tanya is misspelled in the 3rd record! How could this be detected in thousands of records. By using a flat file structure, we create:

#### Data duplication

Duplicate copies of data – we would have to key in Tanya Travels Website 3 times. This wastes storage space and means that data is redundant or unnecessarily repeated.

#### Data inconsistency

The more often data has to be keyed the more chance there is for errors in data entry. In this case the data is inconsistent because Tanya is spelt differently in two different entries.

#### Poor data Integrity

Data that is erroneous or inconsistent is not reliable.

### The main features of a relational database

Relational database structures allow data to be stored across a number of tables. The tables are linked or related together, using relationships, on common fields. A relational database is a much more powerful way of storing data as the relationships allow data to be retrieved from a number of tables using a single query and there is less data redundancy.

- Each table must have a unique reference for each record which is called the **Primary key**. The primary key uniquely identifies each record in the table.
- Replicating a Primary Key into other tables creates a **Foreign key**.
- Foreign keys form the **Relationships** that link the tables together.

In a database design that applies the rules of the relational model, each data item, a Project Name for example, is stored only once, in one location. This avoids having to maintain the same data in multiple locations. The duplication of data is called data redundancy and this is kept to a minimum in relational databases.

#### Tables

A relational database has more than one table of data. The tables are joined through primary and foreign keys. The key can join tables together in many different relationships, such as a one-to-one relationship, where each table contains one unique record; a one-to-many relationship, where one table might contain records about an individual and the other might have records about all that individual's financial transactions.

#### Constraining input

Using a relational database a name and data type for each field can be specified. As well as data types, database systems allow you to apply further constraints such as specifying a maximum length or enforcing the uniqueness of a field. Constraints

help control data integrity and prevent problems such as:

- entering a Project Name (text) in a field where a Project Number is expected.
- entering a Start Date in the last century.
- having two projects with the same Project ID.
- entering a number in the Start Date field.
- entering a Pay Grade which does not exist.
- Maintaining data integrity.

The features which allow us to set field properties, link tables and set constraints help to increase the reliability of the data.

### Access Rights

Most relational database systems provide access rights which can be assigned to different users depending on their role. Some of the operations that can be allowed or disallowed to a user are SELECT, INSERT, DELETE, ALTER, And CREATE. These

rights correspond to the operations that can be performed using the Structured Query Language (SQL).

### Structured Query Language (SQL)

When manipulating data in the database, SQL queries are formulated. Examples of manipulating data include storing new data, and selecting and altering existing data.

### Create, read, update and delete

All databases will allow their owners to create, read, update and delete data. This is generally done through various forms of Structured Query Language. The language also allows most users to query and manipulate data, as well as protect it from unwanted updates, deletions or other potential errors.

### Portability

The relational model is a standard. By adhering to the rules of the relational model you ensure that your data can be transferred between relational database systems relatively easily.

### Key Terms Associated with Relational Databases

Term	Definition
Attribute	An individual data item within an entity. For example Employee Name in an entity called EMPLOYEE. This is also known as a field.
Entity	Identifies an object about which data will be stored within the system. An entity could be described as a person, place, thing, or event which has relevance to the database. For example an EMPLOYEE or a PROJECT. Another name for an entity is a table in the relational model. Note when describing an entity, singular names are used. An entity occurrence is an instance of the entity. For example "10, Margaret Lynch" is an entity occurrence in the EMPLOYEE entity. This is the same as a record.
Primary key	A primary key consists of one or more attributes that identifies one record from any other. For example the Employee No would uniquely identify each employee in a table.
Composite key	Consists of more than one attribute to uniquely identify an entity occurrence or record.
Foreign key	A foreign key is a primary key in one entity which is included in another entity to link the two tables.
Relationships	A relationship is an association of entities. A relationship is established by a foreign key in one entity linking to the primary key in another. For example, an employee works in a department. A department code could be included as a foreign key in the employee entity to link to the department entity. The degree or cardinality of a relationship is the number of occurrences in one entity which are associated to the number of occurrences in another. The degree of relationship can be: <ul style="list-style-type: none"> <li>• One-to-One (1:1)</li> <li>• One-to-Many (1:M)</li> <li>• Many-to-Many (M:N)</li> </ul> Cardinality shows the maximum number of occurrences that can be involved in a relationship [.

Referential integrity	<p>Referential integrity ensures that the data in one table does not contradict the data in another table. Specifically, every foreign key value which exists in a table must have a matching primary key value in a related table. This is used to manage the relationships between primary and foreign keys.</p> <p>When entering a new employee, the Pay Grade which they are allocated needs to be specified. Pay Grade will be a foreign key in the employee table and the primary key in the pay table. In order to preserve the integrity of the data in the database there rules that need to be observed:</p> <ul style="list-style-type: none"> <li>• If inserting an employee in the employee table, the insertion should only be allowed if their Pay Grade exists in the pay table.</li> <li>• If deleting or changing a Pay Grade in the pay table, the deletion or update should only be allowed if there are no employees on that Pay Grade.</li> <li>• If changing the value of a Pay Grade in the employee table, the update should only be allowed if the new value exists in the Pay table.</li> </ul>
-----------------------	--

**Data Modelling**

Database design is about identifying the relationships between data and creating a logical design which is best suited to the data required for a system.

**E-R Models**

Data modelling is used when designing relational databases. Within the process of data modelling, Entity Relationship (ER) diagrams are produced, and plans for the table design are constructed.

The first step in designing a database is to identify and state what data needs to be stored. From the statement of data requirements a conceptual data model is produced. This describes how the data elements in the database are to be grouped. The terms used in building a picture of the data requirements are Entity, Attribute and Relationship. Following this, the ENTITY-RELATIONSHIP (ER) diagrams are produced.

**Logical Data Models**

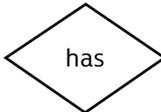
When creating a logical data model entities and their relationships are identified with the keys. The attributes of each entity are also identified. A logical data model does not describe the physical structure of the real database.

**Physical Data Models**

A physical data model is derived from the logical data model and includes the specification of all tables and the columns (fields) inside them. The table specification includes the table name and the column specification includes the column name and data type. The physical data model contains the primary keys of each table and it specifies the

relationship between the tables using foreign keys. The physical implementation will differ based on the actual database management system used.

**E-R Diagram Notation and Symbols**

Symbol	Meaning
	Entity : Represented by a rectangle for example EMPLOYEE
	Relationships represented by a diamond
	Attribute

When drawing an ER diagram:

1. Create a box for each entity relevant to the model.
2. Model the relationships between each entity by drawing lines to connect related entities.
3. Label the relationships using verbs.
4. Use Crows Foot Notation to illustrate the cardinality of relationships.
5. Identify relevant attributes within important entities.
6. Note that many to many relationships cannot be implemented in a relational database so these should be resolved prior to developing the physical data model.

This will help define how different business concepts relate to one another, and created a solid conceptual foundation for designing a relational database to support the business requirements.

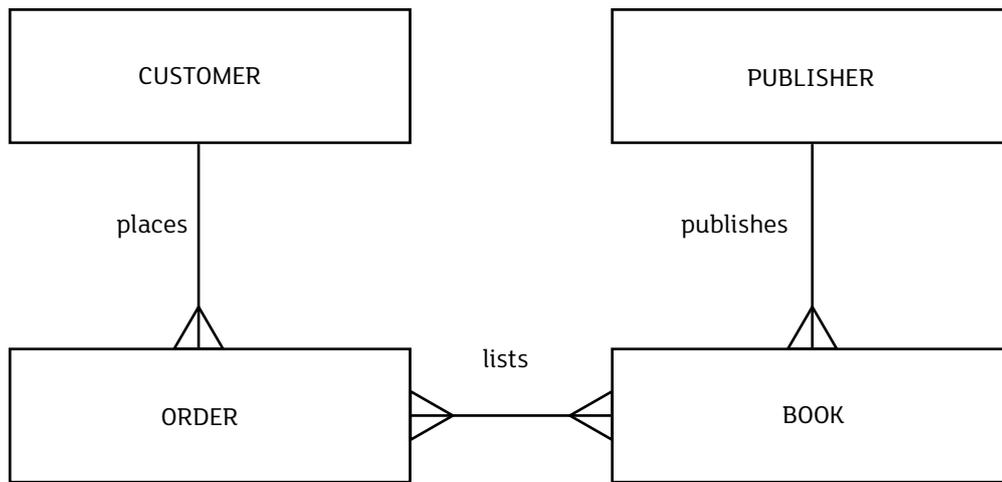
When designing databases you should not have any 'many to many' relationships. They need to be redesigned so that they are 'one to many' or 'many to one'.

Consider the scenario: A company uses a database to store data about orders.

- Customers can make multiple orders
- Each order can refer to a number of books
- A book has only one publisher

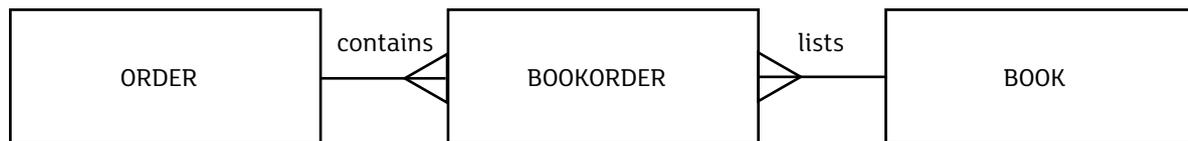
Draw an entity relationship (ER) diagram for this system which does not contain a many to many relationship.

Use the instructions outlined above to construct the diagram.



Note many-to-many relationships cannot be implemented in a relational database. An order can list many books and a book can be listed on many orders.

This can be changed to the relationship below:



The many-to-many relationship is now resolved. The E-R Diagram now becomes (with associated relationships):

