FACTFILE:

GCE DIGITAL TECHNOLOGY

UNIT A2 1: INFORMATION SYSTEMS

Rewarding Learning

Databases 2

Learning Outcomes

Students should be able to:

- describe the characteristics of data in un-normalised form, first normal form(1NF), second normal form(2NF) and third normal form(3NF)
- describe the advantages and disadvantages of normalisation
- normalise data requirements to 3NF

Content in Program Structure

- The characteristics of data in un-normalised form
- First normal form,(1NF), second normal form(2NF), third normal form(3NF)
- Normalise data requirements to third normal form
- The advantages and disadvantages of normalisation
- Describe the components of a data dictionary.

The characteristics of data in un-normalised form

Consider the data table below

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	**Tanye** 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

The table can also be presented as follows:

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

Creative Web Designers works on designing Websites for clients. Above you can see a section of data about employees and the different projects that they work on in the company. Employees are given a pay rate based on their experience. For each project that they work on, they are given an allocation of hours. This is used to calculate the amount of money to be paid at the end of the project.

Task: Draw an ER diagram to represent the data in the system

The data is presented as a flat file structure and as such is in un-normalised form. Un-normalised data is characterised by redundancy and inconsistencies. There are repeated fields and multiple values of attributes (Employee No, Name, Start Date, Pay Grade, Hourly Rate, Hours Allocated) for one Project. Here, one project can use several employees and that data will be reflected in a single row in the database. Additionally, Margaret Lynch is being stored twice which results in a lot of memory consumption and data redundancy. Under these circumstances, anomalies can arise in the data. For example, if Margaret Lynch was to move to Pay Category B, two updates would be required in the flat file table. If the data is updated only in one place, then the content of the file is incorrect leading to poor data integrity. This problem has been caused by an update anomaly. When data has high levels of redundancy anomalies will arise, integrity is low and there will be inconsistencies due to data duplication. The data in the table above could be normalised to enable it to be implemented in a relational database.

Normalisation is a formal bottom up process. The documents used within an organisation can be examined. Data to be stored can be identified from within the documents. Normalisation involves the application of a set of rules to remove undesirable characteristics and problems and can be used to test the correctness of an ER model. Normalisation will reduce data redundancy, storage space, and update anomalies such as insertion, deletion and modification. It improves the integrity of the data.

Un-normalised form - UNF

To write data in UNF complete each of the following steps:

- √ Select a name for the main entity – in this case PROJECT.
- √ Each field contained within the entity is listed in brackets.
- √ Primary keys are underlined.
- √ Foreign keys are represented with *.
- √ Repeating groups are contained within {}.
- √ Derived attributes such as average Hourly Rate should not be included here but the developer may decide to store the value to improve performance.

The Project data can be written in un-normalised form as follows:

PROJECT(Project No, Project Name, {Employee No, Name, Start Date, Pay Grade, Hourly Rate, Hours allocated})

Repeating groups

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
		12	Gerry Mattwell	09/07/16	B	£80	20
		19	Ruth Debson	20/08/14	C	£70	30
1902	Action Pursuits Website	10	Margaret Lynch	30/03/16	A	£90	30
		18	Alex Kay	01/06/15	B	£70	25

Project Table in un-normalised form

First Normal Form(1NF)

Data is said to be in 1NF if all attributes are atomic and entities do not have repeating groups. (Atomic means that the attribute cannot be further sub divided).

Clearly the table violates the rules for 1NF as there are repeating groups identified. We must remove the repeating groups into a new table with a copy of the primary key from the main table. Call the new table (or Entity) PROJECT_EMPLOYEE.

PROJECT_EMPLOYEE (Project No*, Employee No, Name, Start Date, Pay Grade, Hourly Rate)

PROJECT now becomes PROJECT(Project No, Project Name)

Project No	**Employee No**	**Name**	**Start Date**	**Pay Grade**	**Hourly Rate**	**Hours**
1200	10	Margaret Lynch	12/12/15	A	£90	25
1200	12	Gerry Mattwell	09/07/16	B	£80	20
1200	19	Ruth Debson	20/08/14	C	£70	30
1902	10	Margaret Lynch	30/03/16	A	£90	30
1902	18	Alex Kay	01/06/15	B	£70	25

PROJECT_EMPLOYEE Table

Project No	Project Name
1200	Tanya Travels Website
1902	Action Pursuits Website

PROJECT Table

PROJECT Table and PROJECT_EMPLOYEE Table in 1NF

The repeating groups have been removed from the PROJECT table and placed into a new table called PROJECT_EMPLOYEE. This table has a **composite** primary key made up of Project No and Employee No. This means that Project No and Employee No are required to uniquely identify a row in the PROJECT_EMPLOYEE table. For example the Project No 1200 would return 3 rows, whereas Project No 1200 and Employee No 10 would give one unique row.

A copy of the Project No is kept in the PROJECT table. In the PROJECT_EMPLOYEE table, Project No is referred to as a foreign key. This facilitates the link or relationship between the two tables.

Second Normal Form(2NF)

Data is in 2NF if it fulfils the conditions for 1NF and all non-key attributes in the relation are fully dependent on the primary key. Non-key attributes which are only partially dependent on the key are removed to a new entity along with a copy of their determinant (key attribute on which they are partially dependent).

Project No	Project Name
1200	Tanya Travels Website

Since the PROJECT table has a single primary key field, there can be no partial key dependencies. A table with a single attribute as a primary key is in 2NF.

The PROJECT_EMPLOYEE table has a composite key made up of two attributes. In this case each non-key attribute is checked for dependency on the key attributes. This table is not in second normal form because:

- Employee No can be used on its own to find one unique Employee Name, Start Date, Pay Grade and Hourly Rate. For example Employee No 10 is always Margaret Lynch. Therefore Employee Name, Start Date, Pay Grade and Hourly Rate are partially dependent on the composite key (Project No and Employee No). That is they are functionally dependent on Employee No only. Employee No is their determinant.
- In order to determine the Hours Allocated however, both the Project No AND Employee No are required. Therefore Hours Allocated is fully functionally dependent on the key and remains in the entity PROJECT_EMPLOYEE.

	Project No	Employee No
Name		√
Start Date		√
Pay Grade		√
Hourly Rate		√
Hours Allocated	√	√

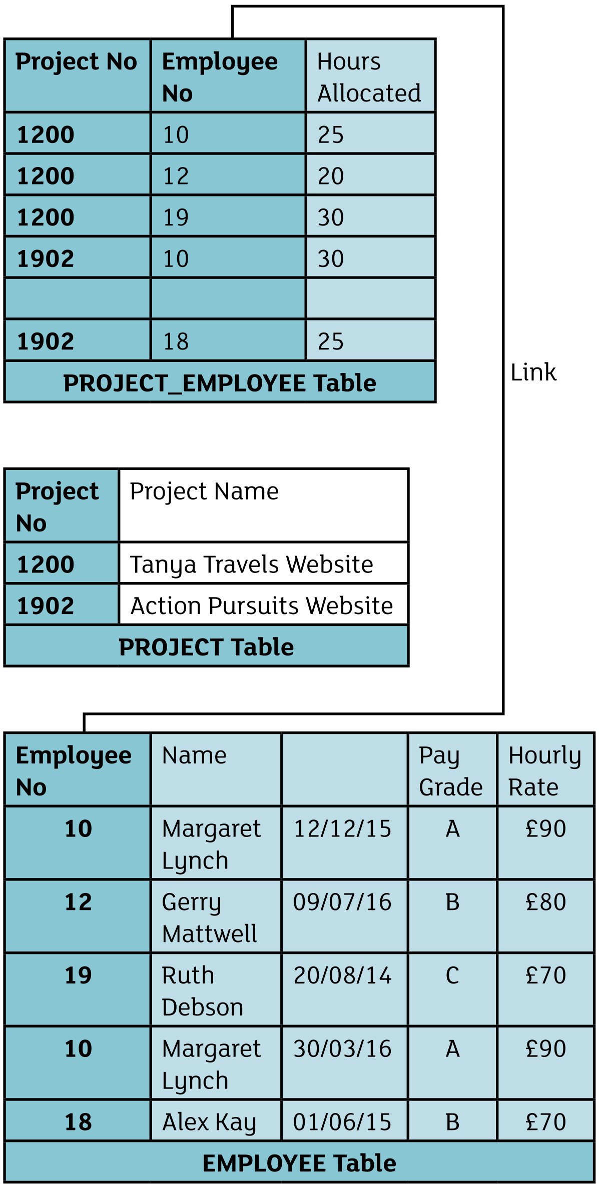
Table showing functional dependencies in the PROJECT_EMPLOYEE Table

The new entity will be called EMPLOYEE.

EMPLOYEE(Employee No, Name, Start Date, Pay Grade, Hourly Rate)

PROJECT_EMPLOYEE now becomes:

PROJECT_EMPLOYEE(Project No*, Employee No*, Hours Allocated)



Project No	**Employee No**	Hours Allocated
1200	10	25
1200	12	20
1200	19	30
1902	10	30
1902	18	25

PROJECT_EMPLOYEE Table

Project No	Project Name
1200	Tanya Travels Website
1902	Action Pursuits Website

PROJECT Table

Employee No	Name		Pay Grade	Hourly Rate
10	Margaret Lynch	12/12/15	A	£90
12	Gerry Mattwell	09/07/16	B	£80
19	Ruth Debson	20/08/14	C	£70
10	Margaret Lynch	30/03/16	A	£90
18	Alex Kay	01/06/15	B	£70

EMPLOYEE Table

Diagram showing the new table EMPLOYEE, the PROJECT_EMPLOYEE table and the relationship between them

Third Normal form (3NF)

Data is in third normal form if it is in second normal form and there are no non-key dependencies. That is as long as an attribute does not depend on another attribute which is not the primary key. This is called a transitive dependency.

All tables must be examined to ensure that there are no transitive dependencies. If a transitive dependency exists the attribute and its determinant must be removed to a new entity. A copy of the determinant is retained in the original table.

Examine the PROJECT table. Since this table has only one attribute along with the primary key, there are no transitive dependencies.

Examine the PROJECT_EMPLOYEE table. We have already established that all elements in this table depend fully on the composite key.

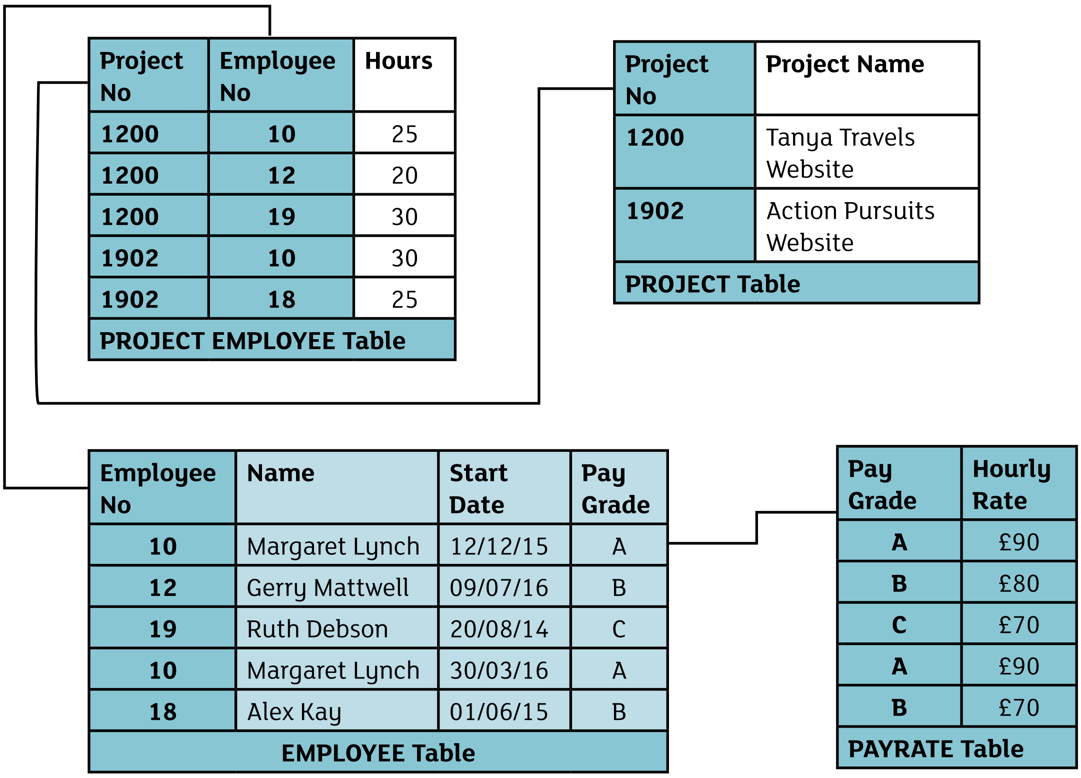
Examine the EMPLOYEE table. Clearly the Pay Grade attribute can be used to find a unique Hourly Rate. This constitutes a transitive dependency. Therefore, Hourly Rate must be removed to a new entity PAYRATE along with a copy of its determinant. The determinant becomes the key field in the new table.

PAYRATE (Pay Grade, Hourly Grade)

EMPLOYEE now becomes

EMPLOYEE(Employee No, Name, Start Date, Pay Grade*)

Note that a copy of the determinant is left in the EMPLOYEE table as a foreign key.



Project No	Employee No	Hours
1200	10	25
1200	12	20
1200	19	30
1902	10	30
1902	18	25

PROJECT EMPLOYEE Table

Project No	Project Name
1200	Tanya Travels Website
1902	Action Pursuits Website

PROJECT Table

Employee No	Name	Start Date	Pay Grade
10	Margaret Lynch	12/12/15	A
12	Gerry Mattwell	09/07/16	B
19	Ruth Debson	20/08/14	C
10	Margaret Lynch	30/03/16	A
18	Alex Kay	01/06/15	B

EMPLOYEE Table

Pay Grade	Hourly Rate
A	£90
B	£80
C	£70
A	£90
B	£70

PAYRATE Table

A Diagram showing the data tables in 3NF and the links between them

Task : Can you insert the cardinality associated with each relationship? How does the final database structure compare with your initial ER Diagram?

Advantages of normalising data

- The resulting database will take up less storage space because the duplication of data is minimised.
- Information retrieval will be more efficient because data is structured effectively. Queries will be processed faster.
- Less redundancy means less inconsistencies in data because data will only have to be entered once.
- Data integrity will be increased.

Disadvantages of normalisation

- Normalisation is a complex process required to create the database structure.
- Normalisation can generate more tables than an un-normalised database
- More tables mean a more complex database and queries may be slow to run.
- It is necessary to assign more relationships to interact with larger numbers of tables
- With more tables, setting up queries can become more complex

The contents of a data dictionary

A data dictionary is a file containing descriptions of the structure of the data stored in a database. The data dictionary contains field names, data types, field size, format, default values, entity or table names and whether or not the field is a primary key and details about validation. A data dictionary can be used to provide metadata or 'data about data'. This allows identifiers to be attached to key data which can then be structured, retrieved and used.