Lawn & Order Tutorial

Purpose and structure of a relational DBMS

Role of primary and foreign keys in a relational DBMS

Dexter owns a business named Lawn and Order. The business specialises in providing lawn and gardening services. Currently, the business employs 12 employees. Each employee works in one of three departments – lawn mowing, gardening, or landscaping.

A flat-file database table showing the details of the employees is shown below.

tblEmployees

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EmployeeID** | **First** | **Last** | **Department** | **Manager** | **Phone** |
| 1 | Charlotte | Nguyen | Lawn mowing | Harry | #518 |
| 2 | Oliver | Williams | Gardening | William | #444 |
| 3 | Olivia | Lee | Landscaping | George | #123 |
| 4 | Jack | Johnson | Gardening | William | #444 |
| 5 | Mia | Wang | Lawn mowing | Harry | #518 |
| 6 | Noah | Anderson | Gardening | William | #444 |
| 7 | Sofia | Harris | Landscaping | George | #123 |
| 8 | Ethan | Nguyen | Lawn mowing | Harry | #518 |
| 9 | Chloe | Patel | Landscaping | George | #123 |
| 10 | Kai | Lin | Landscaping | George | #123 |
| 11 | Mia | Kumar | Gardening | William | #444 |
| 12 | Arjun | Singh | Lawn mowing | Harry | #518 |

**Data redundancy**

Some pieces of data in the flat-file database are stored in more than one location. For example, the Manager Harry is stored in four different locations.

This is known as data redundancy.

**Relational database**

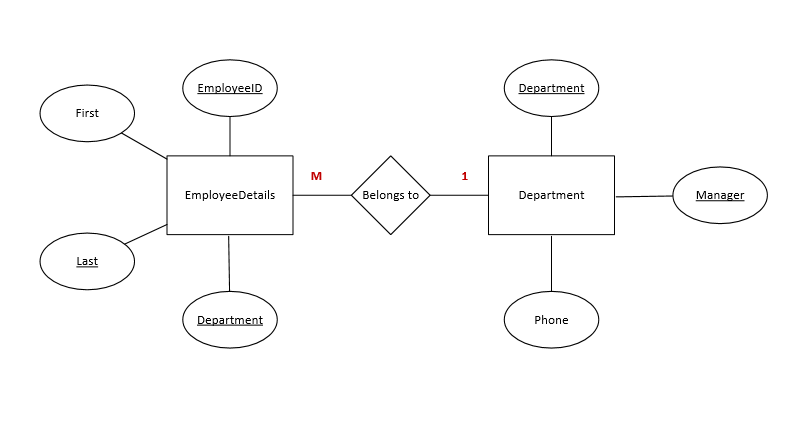
A relational database consists of two or more tables. The data within each table can then be linked together using relationships.

One advantage of a relational database is that it reduces the amount of redundant data stored in the DBMS.

To reduce the amount of data redundancy (or data duplicated) in the tblEmployees the table can be broken down into two tables – tblEmployeeDetails and tblDepartments.

**Entity-relationship (ER) diagram**

An entity- relationship diagram is a graphical representation of the structure of a relational database.



The three main symbols used in an entity-relationship (ER) diagram are:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | Entity | An entity is an object or concept about which a collection of data will be stored. Entities are referred to as **tables** in DBMS software. |
|  | Attribute | An attribute is a single piece of data that makes up an entity. Attribute is another name for a **field**. |
|  | Relationship | A relationship is an association or link between two entities. |

Other symbols used in an entity-relationship (ER) diagram include:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | Connector | Connects symbols within the diagram. |
|  | Key attribute | The key attribute of an entity, or primary key of a table. |
|  | Cardinality | Cardinality refers to the type of relationship between the entities. |

**Required**

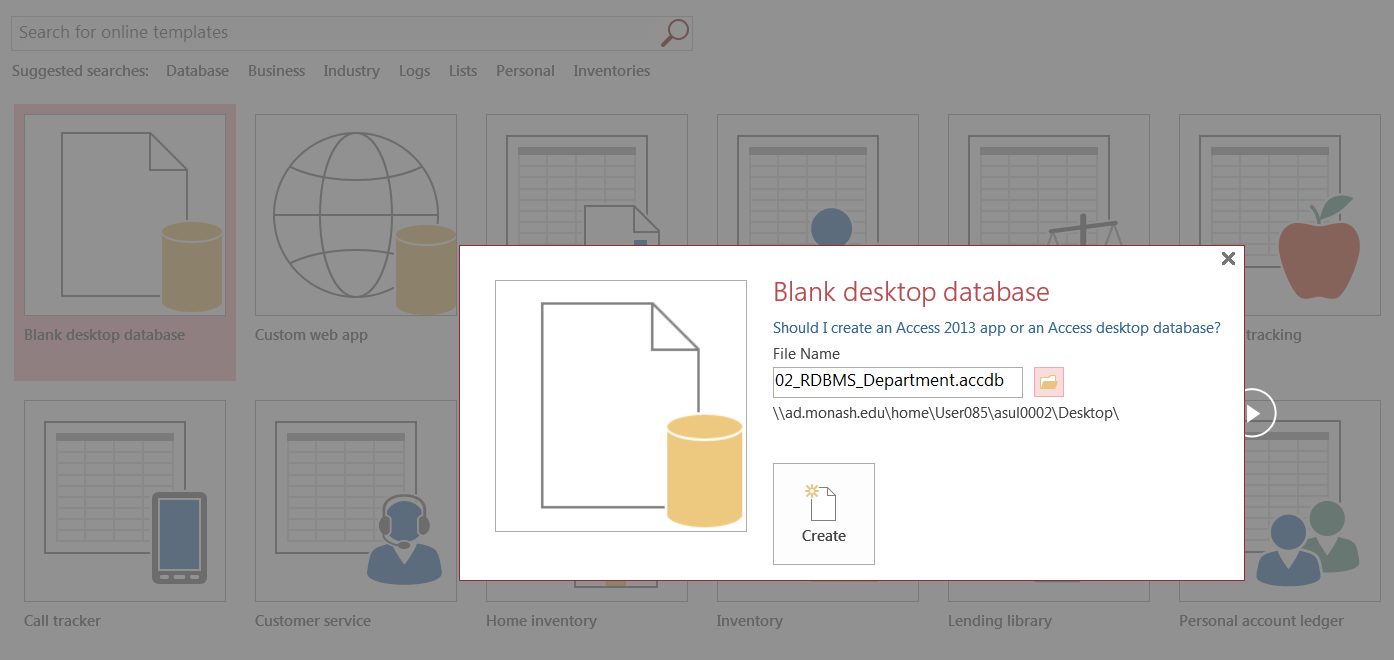
**1. Count** the number of pieces of data in the flat-file database.

**2. Circle** (or highlight) any data that is repeated in the table.

**3. Open** DBMS software.

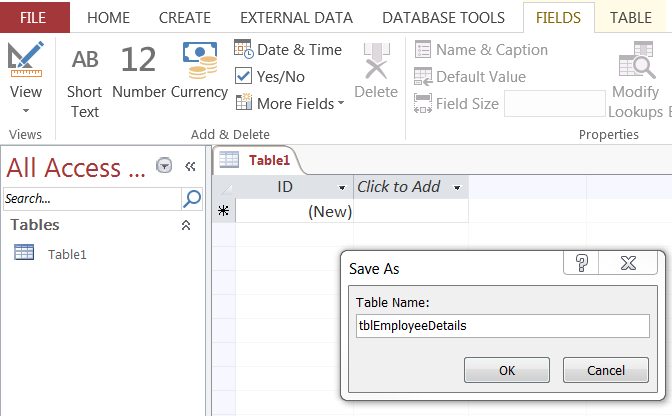
**4. Create** a new blank desktop database.

**5. Name** the database 02\_RDBMS\_Department.



**6. Create** a new table and change to design view.

**7. Name** the table tblEmployeeDetails.

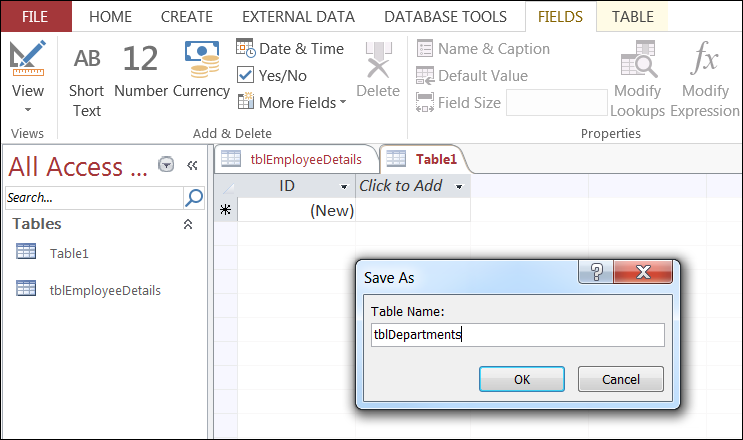


**8. Create** the design of the table based on the data dictionary shown below.

tblEmployeeDetails

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data type** | **Field Size**  **(if short text)** | **Other properties** |
| EmployeeID | Number |  | Primary key |
| First | Short Text | 50 |  |
| Last | Short Text | 50 |  |
| Department | Short Text | 15 | Foreign key |

**9. Create** a new table and name the table tblDepartments.

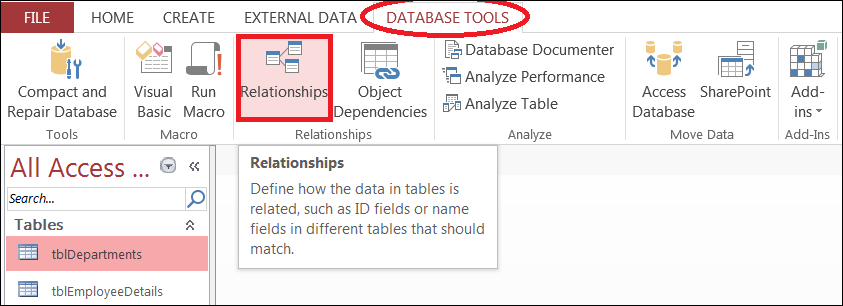


**10. Create** the design of the table based on the data dictionary shown below.

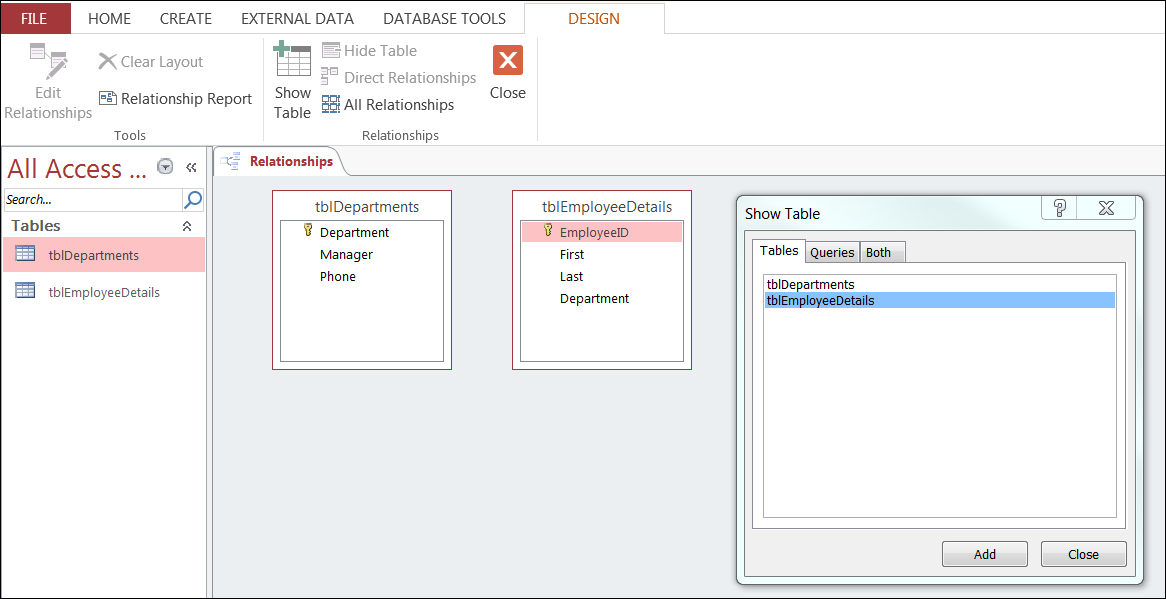
tblDepartments

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data type** | **Field Size**  **(if short text)** | **Other properties** |
| Department | Short Text | 15 | Primary key |
| Manager | Short Text | 50 |  |
| Phone | Short Text | 10 |  |

**11. Select** Database Tools then Relationships.

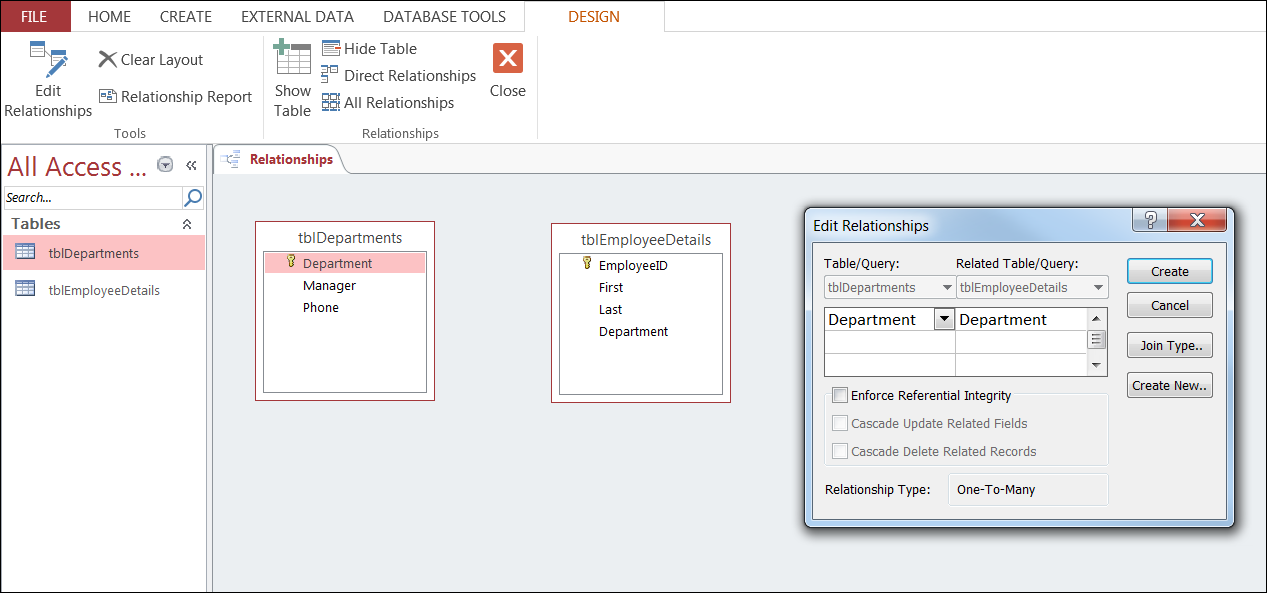


**12. Select** both tables from the Show Table dialog box.

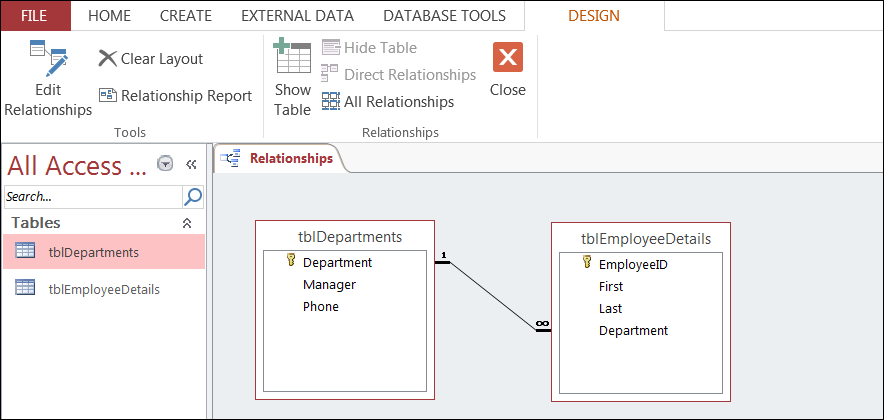


**13. Click** on Department in tblDepartments, **hold down** the mouse and then **drag** across to Department in tblDepartments.

The edit relationship dialog box should appear as shown below.



**14. Select** ‘Enforce Referential Integrity’ checkbox and **click** create.

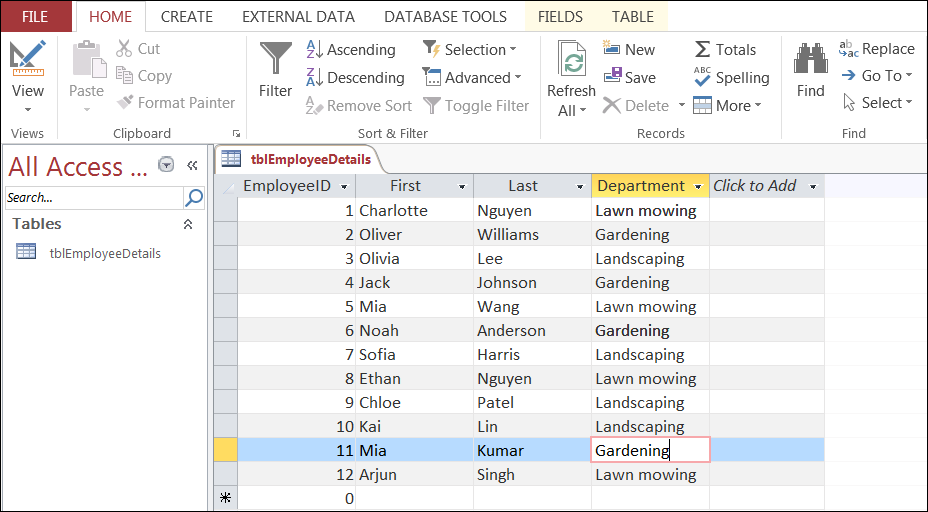


There is now a relationship between the departments and employee details tables.

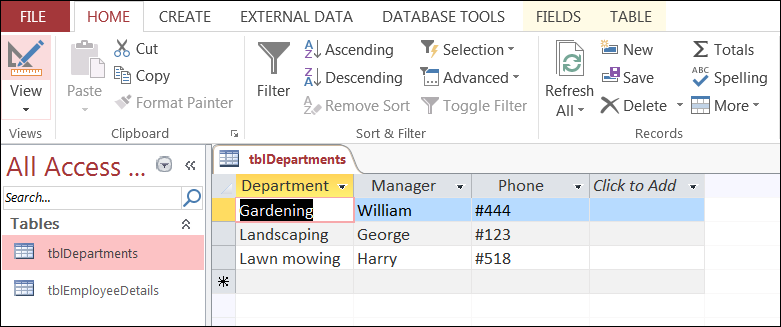
The link or connection has been created using the Department field – as this is common to both tables.

The cardinality between the tables is 1 to many. In this relational database one department can have many employees.

**15. Populate** the table with the details of each employee.



**16. Populate** the table with the department data.



**17. Count** the number of pieces of data stored in the relational database.

**18. Circle** (or highlight) any value that is stored in more than one location in the relational database.