Data Modeling 101

Tables – Columns - Relationships

By Clark Anderson

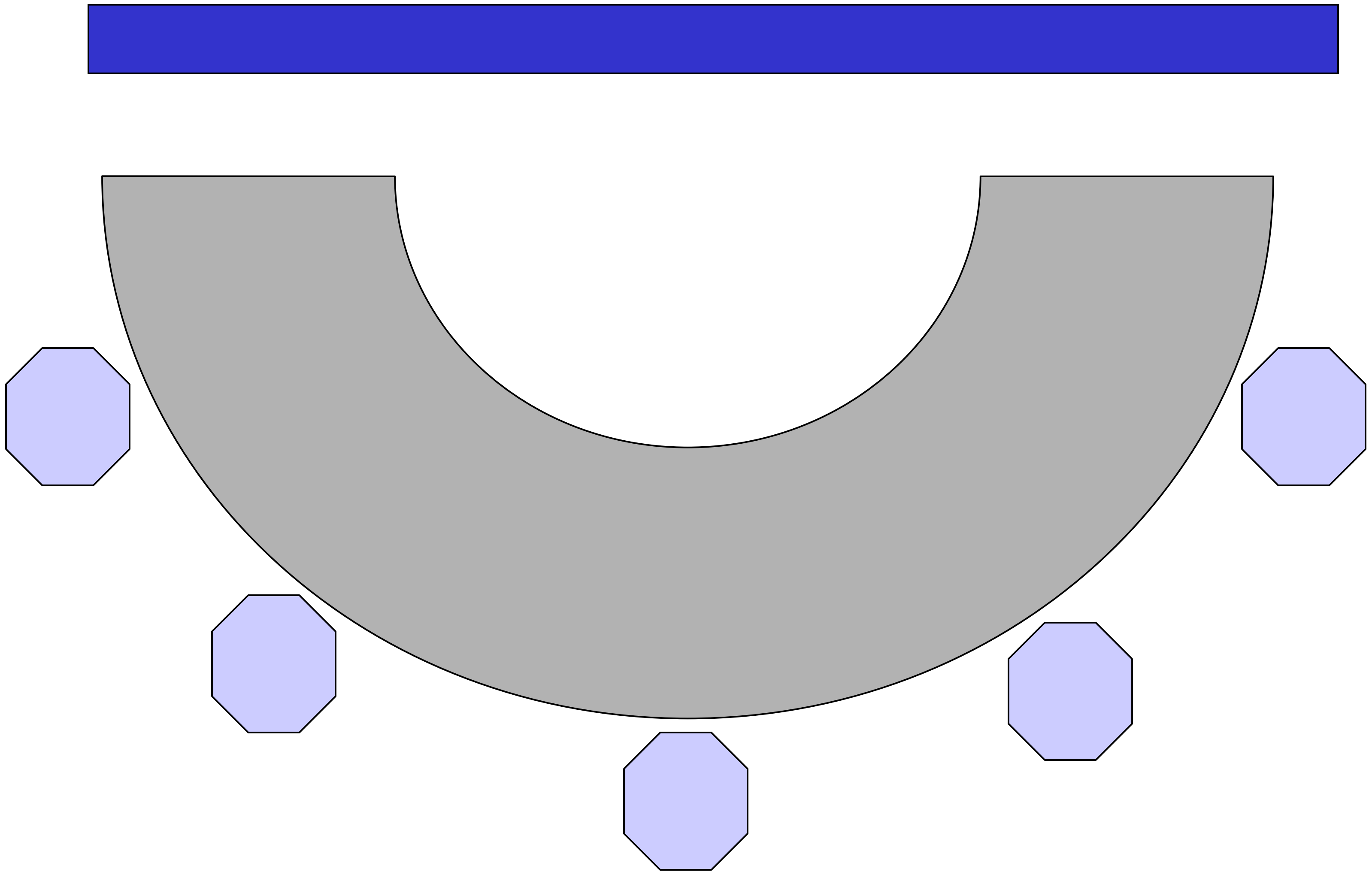
A small software development company is having some difficulties managing the hours spent on the variety of projects.

The CFO has suggested creating a database for this purpose. They hired a consultant to facilitate this data modeling process.

The consultant set up a conference for all of the people involved:

* The CFO, it was her idea,
* the President, he is paying for it,
* the Bookkeeper, he has to run the reports,
* the Project Manager, she has to budget the hours and
* the Programmer, he has to record his hours spent.

The conference room was set up with a projection screen and individual terminals for each attendee to privately enter their input. This arrangement filters out any political pressures. When they are all done, each has had their input, can feel ownership and will enthusiastically work with the end result.



Important Entities (Tables)

They agreed on several important Entities:

1) Position. Each of them served in their own unique position within the organization.

2) Project. These are the major activities for the organization's revenue. They also agreed that this could incorporate the specialized activities also necessary to the organization.

3) Task. It was brought up that most Projects need to be broken down into more manageable Tasks. A given Task may be appropriate for multiple Projects, (especially Software Development Projects).

4) LaborRecord. They needed a place to record the hours spent working on each Task within each Project.

A table was planned for each of these Entities. The consultant was careful to suggest names that described clearly the purpose in the company's everyday language

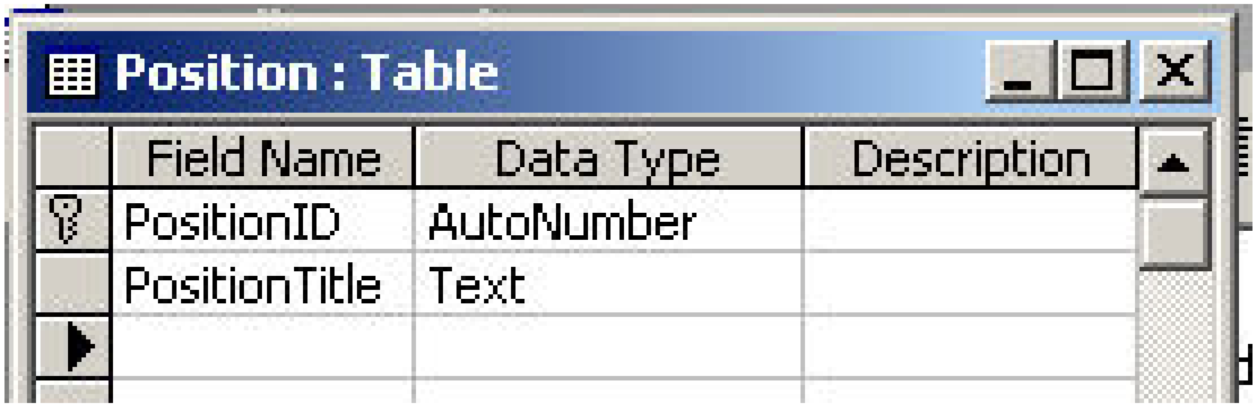
Important Entities (Tables)

* They agreed that the Position Title would be expected to be unique, but there may be reason, in the future, to change it without really changing the meaning and purpose.

* To resolve this, a **system assigned number (AutoNumber)** will be the **Primary Key** and the Position Title will be an Attribute of the Position table.

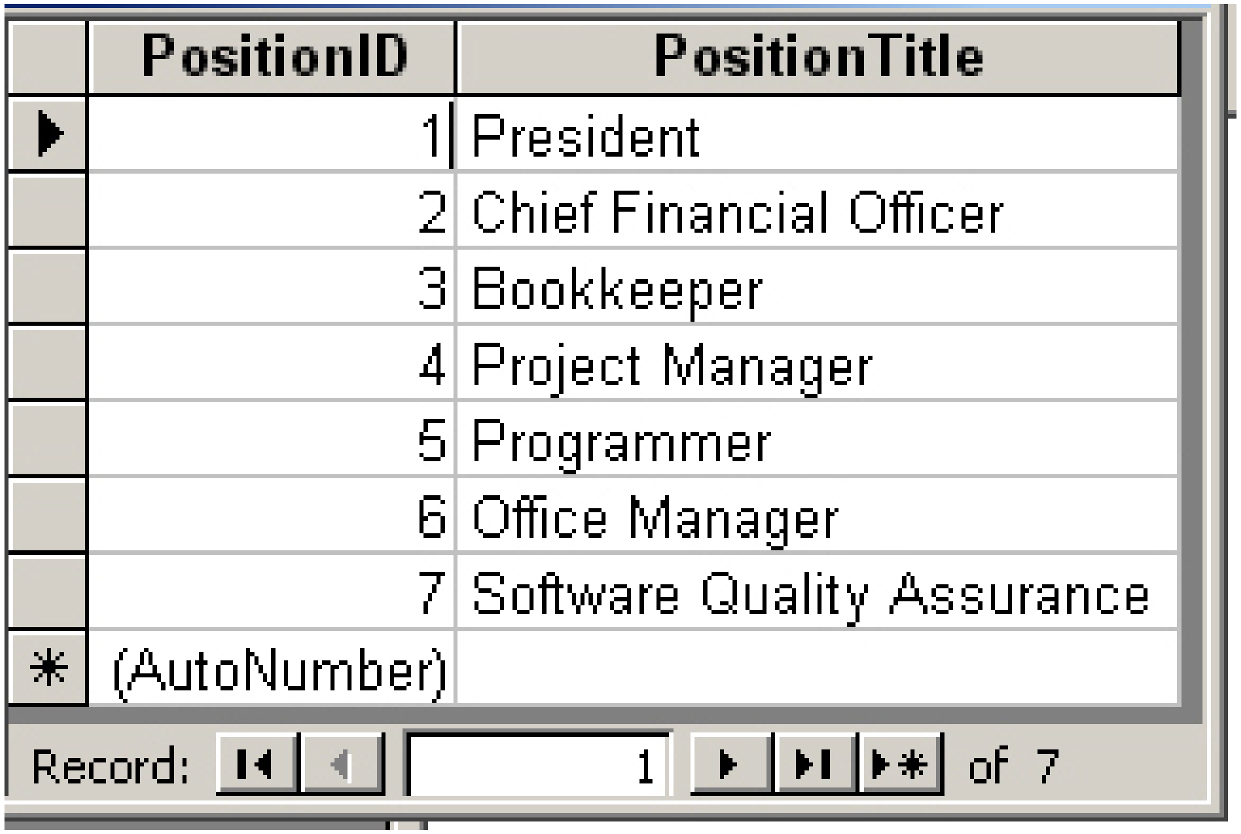
* This same approach made sense for the Project and Task tables.

Important Entities (Tables)



Position : Table

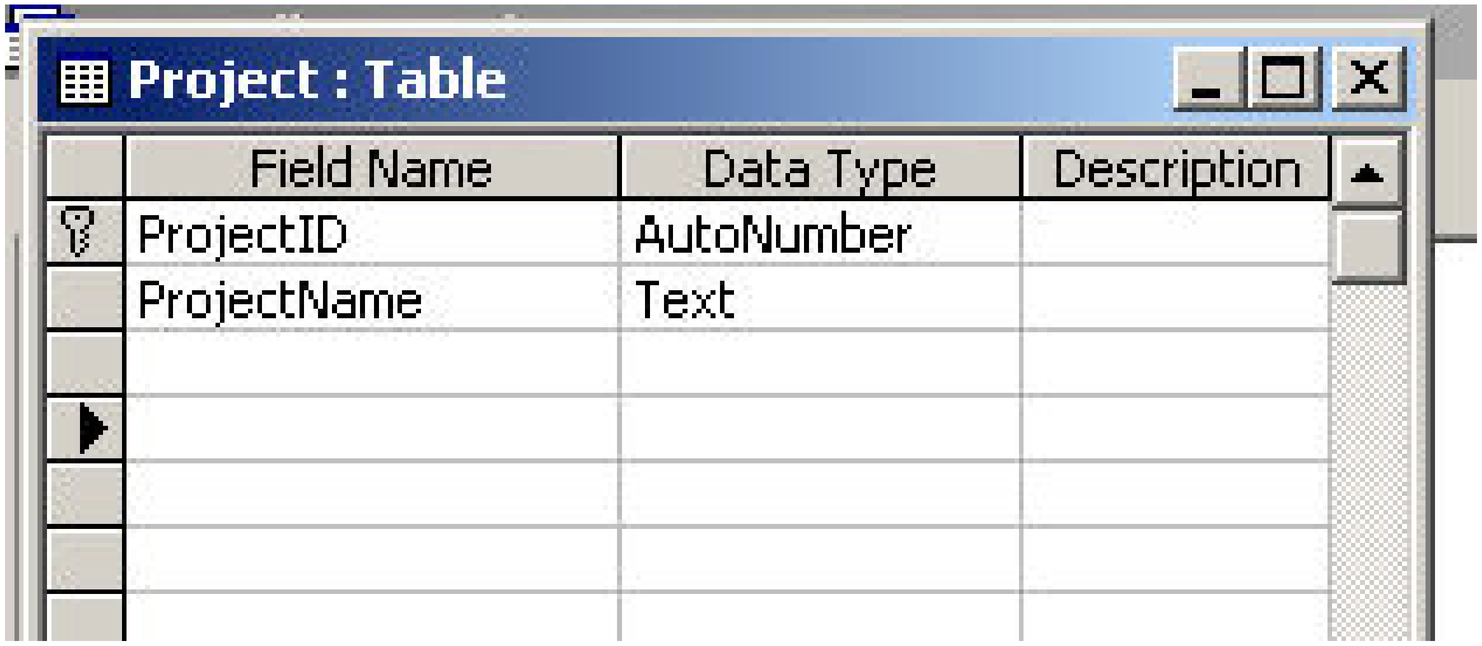
Field Name	Data Type	Description
PositionID	AutoNumber	
PositionTitle	Text	



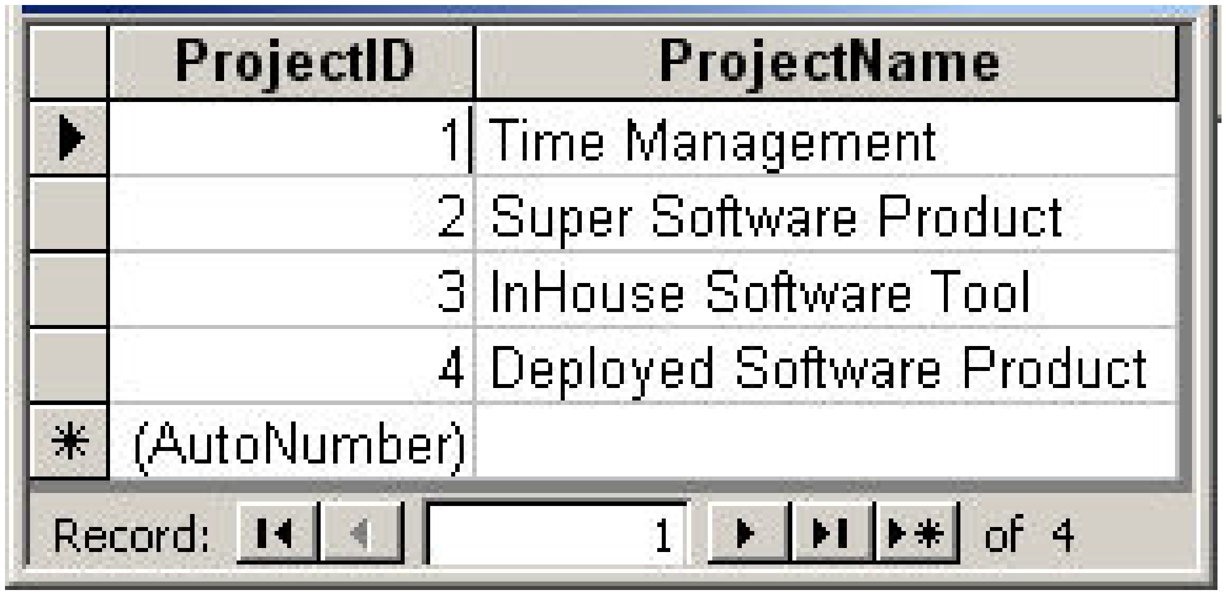
PositionID	PositionTitle
1	President
2	Chief Financial Officer
3	Bookkeeper
4	Project Manager
5	Programmer
6	Office Manager
7	Software Quality Assurance
(AutoNumber)	

Record: 1 of 7

Important Entities (Tables)

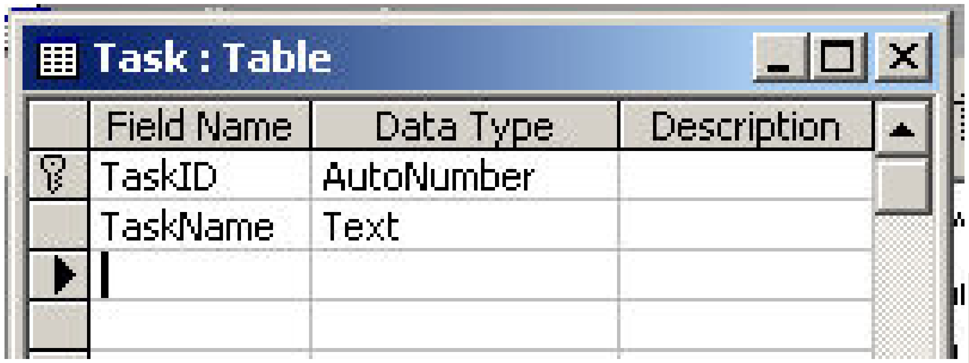


Field Name	Data Type	Description
ProjectID	AutoNumber	
ProjectName	Text	

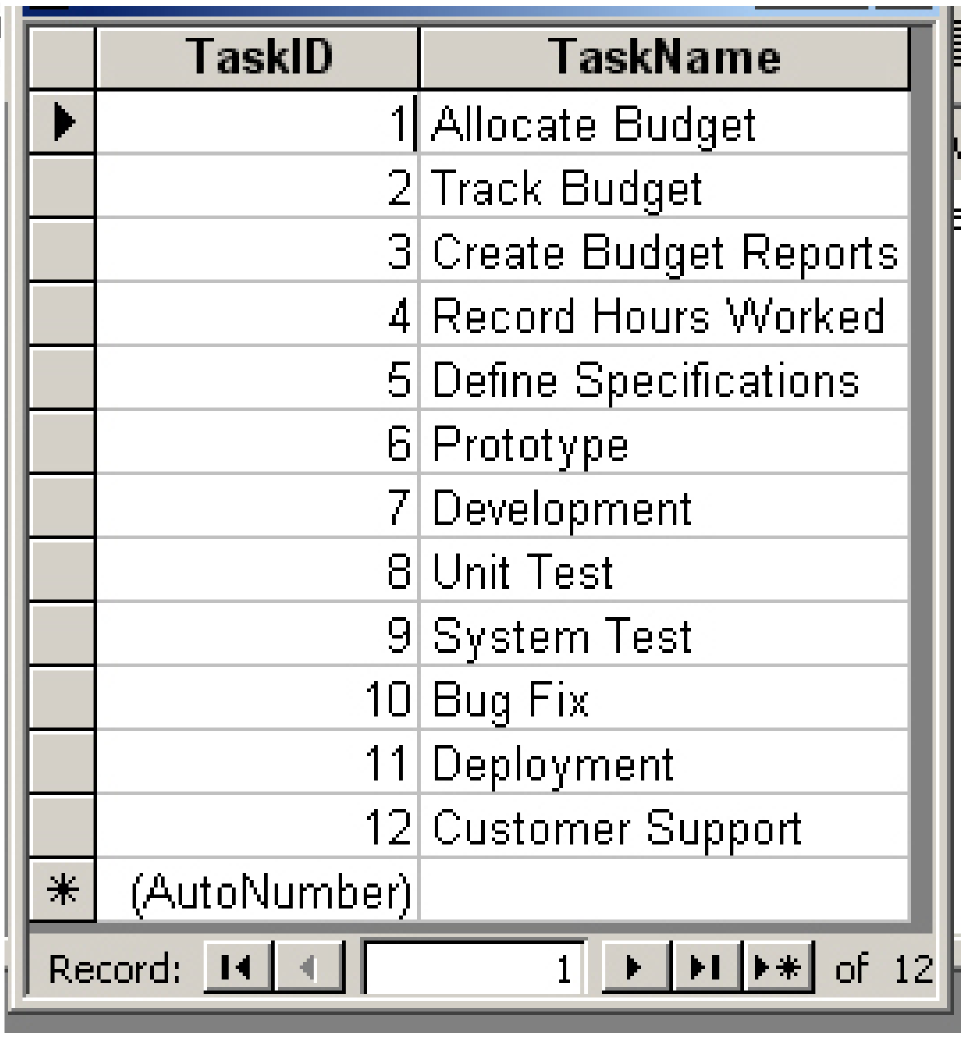


ProjectID	ProjectName
1	Time Management
2	Super Software Product
3	InHouse Software Tool
4	Deployed Software Product
(AutoNumber)	

Important Entities (Tables)



Field Name	Data Type	Description
TaskID	AutoNumber	
TaskName	Text	



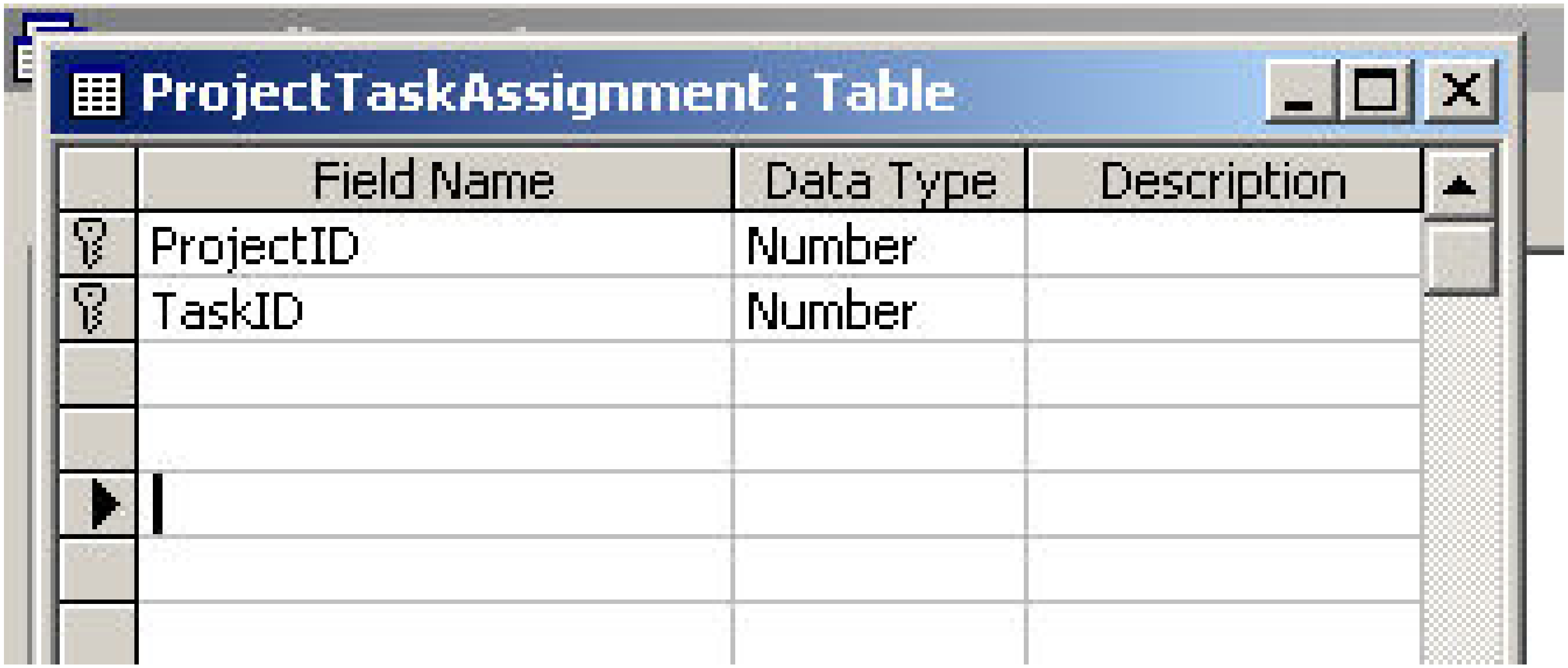
TaskID	TaskName
1	Allocate Budget
2	Track Budget
3	Create Budget Reports
4	Record Hours Worked
5	Define Specifications
6	Prototype
7	Development
8	Unit Test
9	System Test
10	Bug Fix
11	Deployment
12	Customer Support
(AutoNumber)	

The discussions, next, turned to the relations between the tables.

1) A Project may be divided into (be related to) more than one Task.

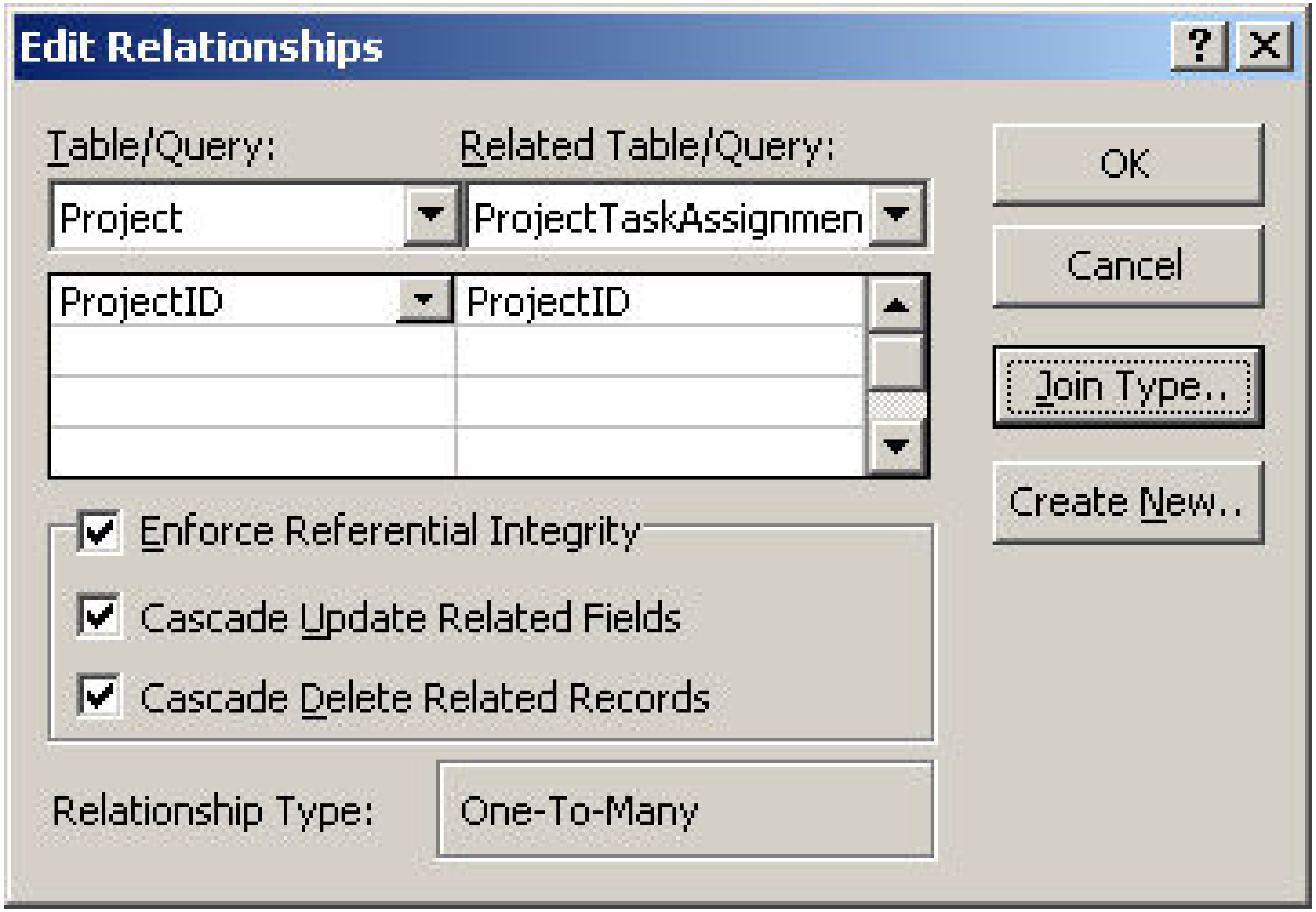
2) A Task may be performed in (related to) more than one Project.

These two yes answers identify a many-to-many relation between Project and Task. To represent this, a **relation table** is created:



Field Name	Data Type	Description
ProjectID	Number	
TaskID	Number	

Example Relationship Edit Forms


Edit Relationships

Table/Query: Project

Related Table/Query: ProjectTaskAssignmen

ProjectID	ProjectID

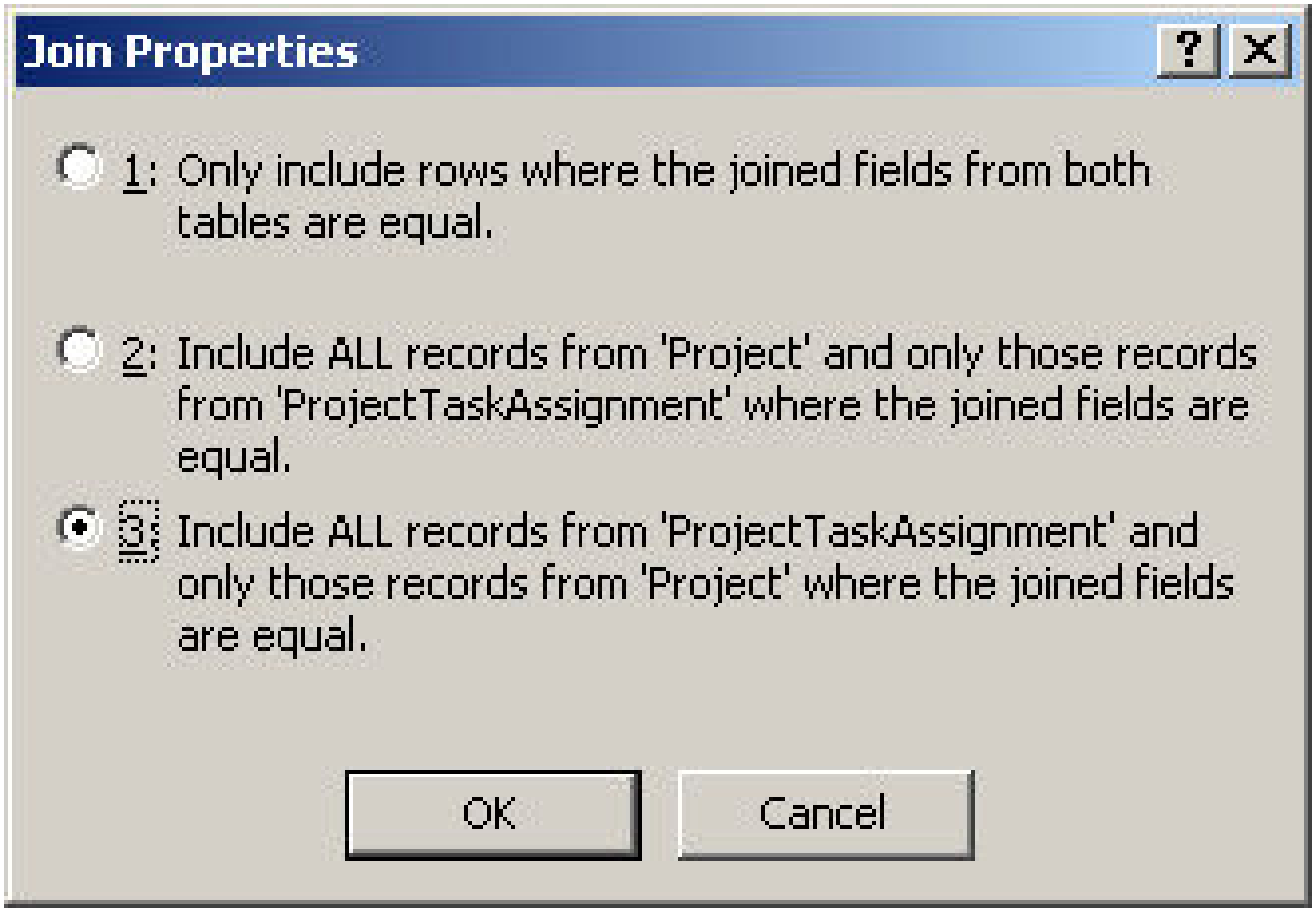
[✓] Enforce Referential Integrity

[✓] Cascade Update Related Fields

[✓] Cascade Delete Related Records

Relationship Type: One-To-Many

Example Relationship Edit Forms



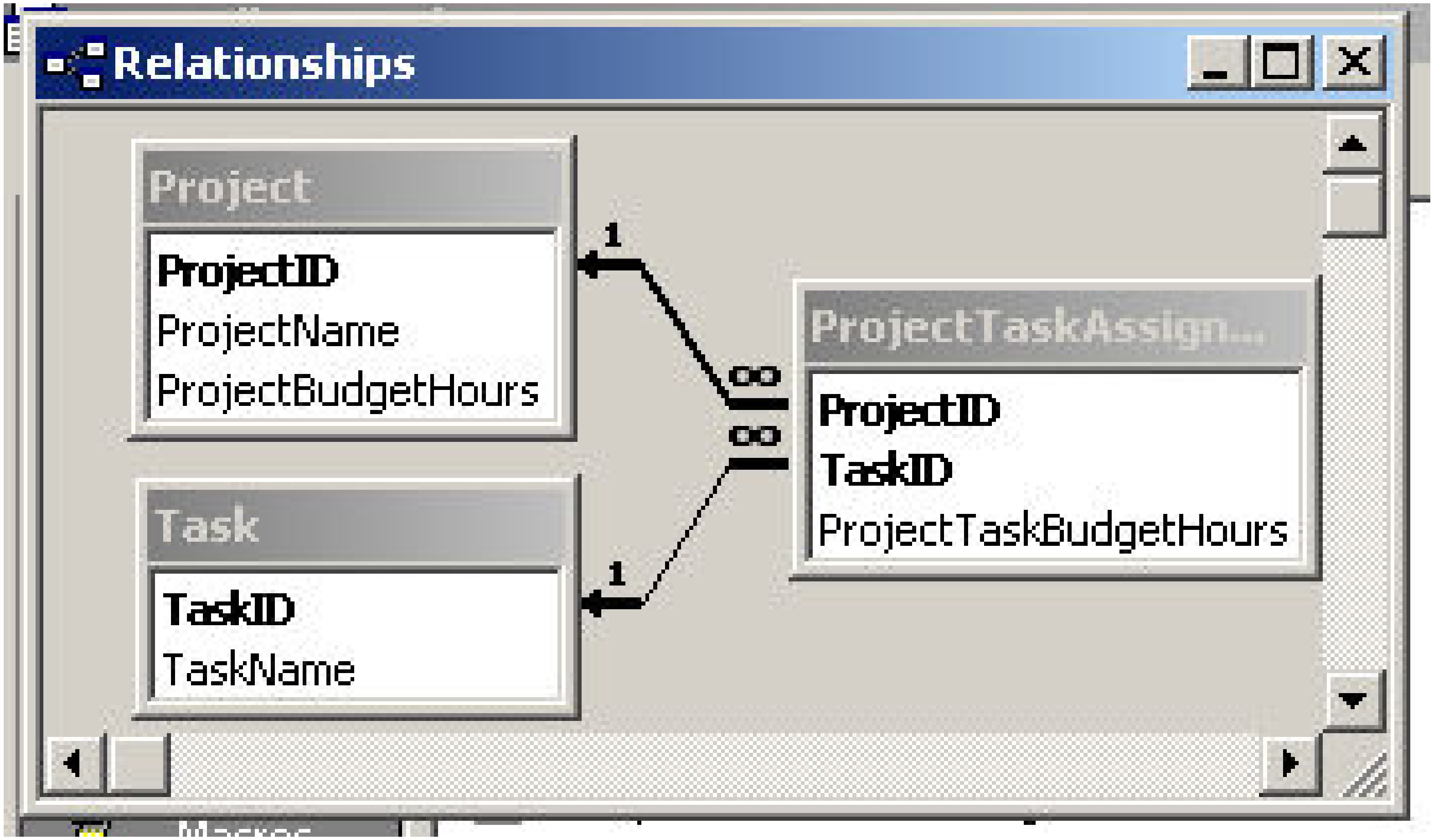
Join Properties

INNER JOIN — 1: Only include rows where the joined fields from both tables are equal.

LEFT JOIN — 2: Include ALL records from 'Project' and only those records from 'ProjectTaskAssignment' where the joined fields are equal.

RIGHT JOIN — [✓ 3: Include ALL records from 'ProjectTaskAssignment' and only those records from 'Project' where the joined fields are equal.]

ProjectTaskAssignment table contains a list of all Tasks performed in each of the Projects. The Primary Keys from Project and Task are brought into the ProjectTaskAssignment table as **Foreign Keys** and combined to form its **Primary Key**.

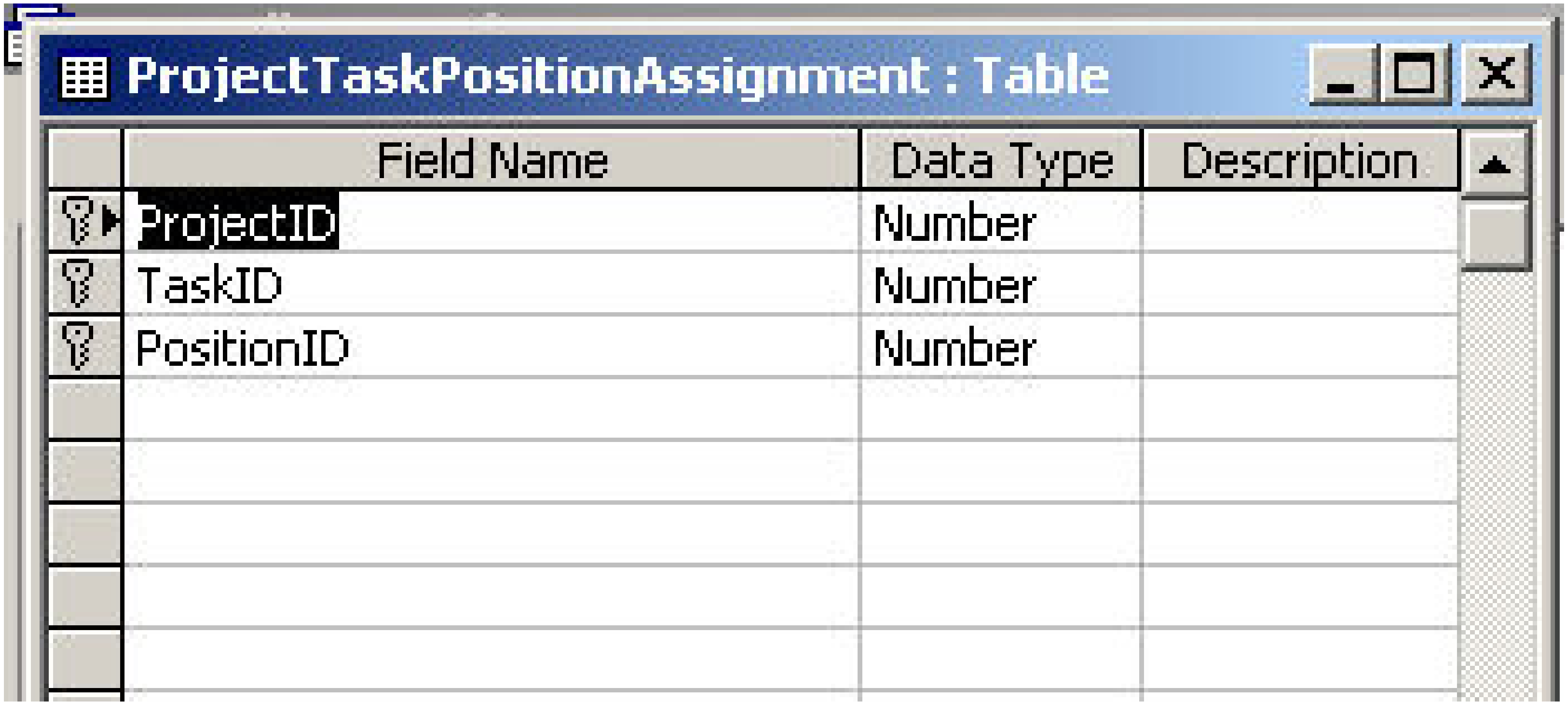


The next relation involves the Position table.

1) A ProjectTaskAssignment may require the labors of (be related to) more than one Position.

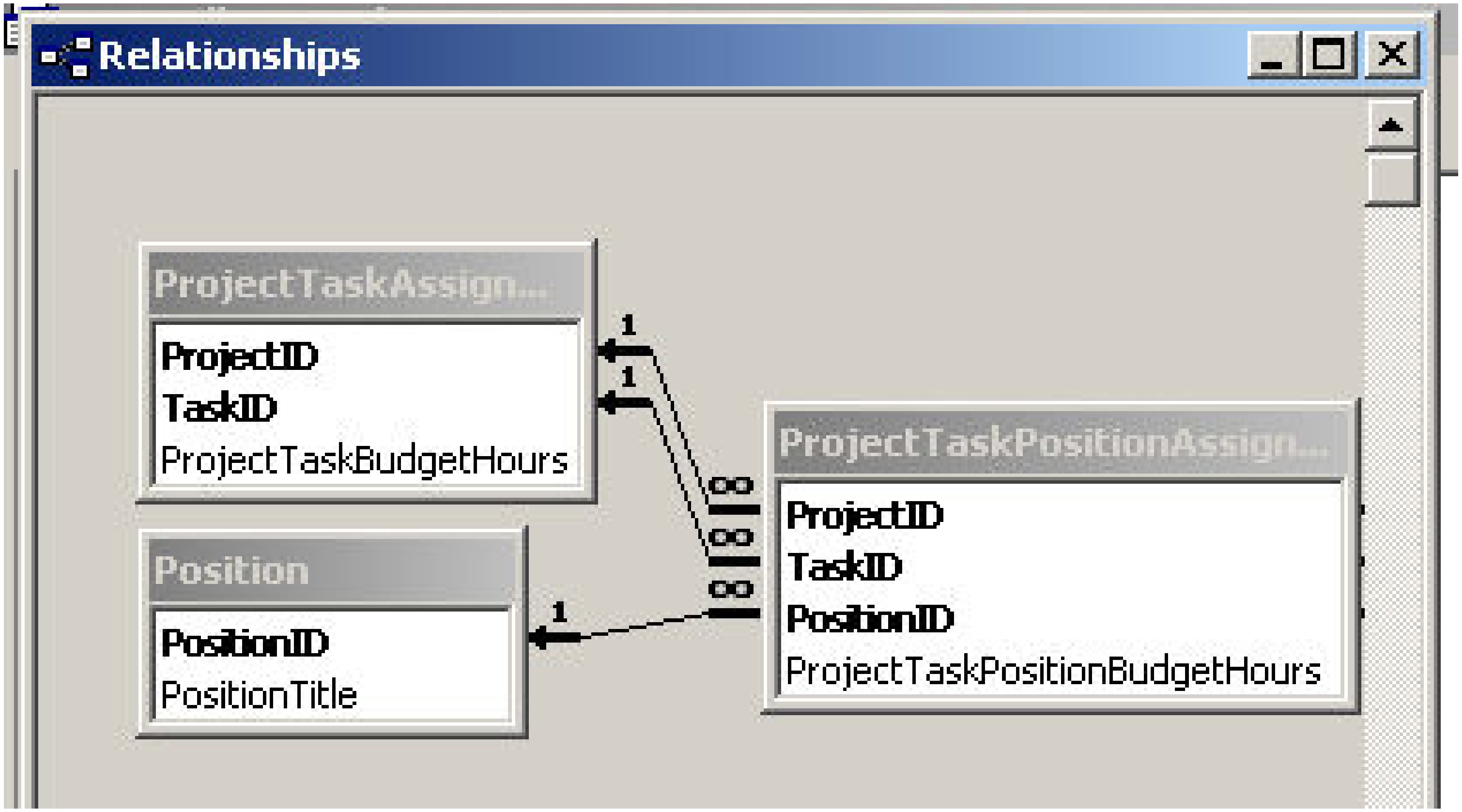
2) An individual Position may be asked to work on (be related to) more than one ProjectTaskAssignment.

Another many-to-many relation is identified. To represent this, another **relation table** is created:


ProjectTaskPositionAssignment : Table

Field Name	Data Type	Description
ProjectID	Number	
TaskID	Number	
PositionID	Number	

ProjectTaskPositionAssignment table contains a record of all Positions assigned to the list of ProjectTaskAssignments. The Primary Keys from ProjectTaskAssignment and Positions are brought into the ProjectTaskPositionAssignment table as **Foreign Keys** and combined to form its **Primary Key**.



Virtual Entity: Time

Every Data Model can have a virtual Entity called Time.

Consider Time as a virtual table with one row for **each point in the time of interest**. (e.g.: **11/28/2012 3:12:03 PM**)

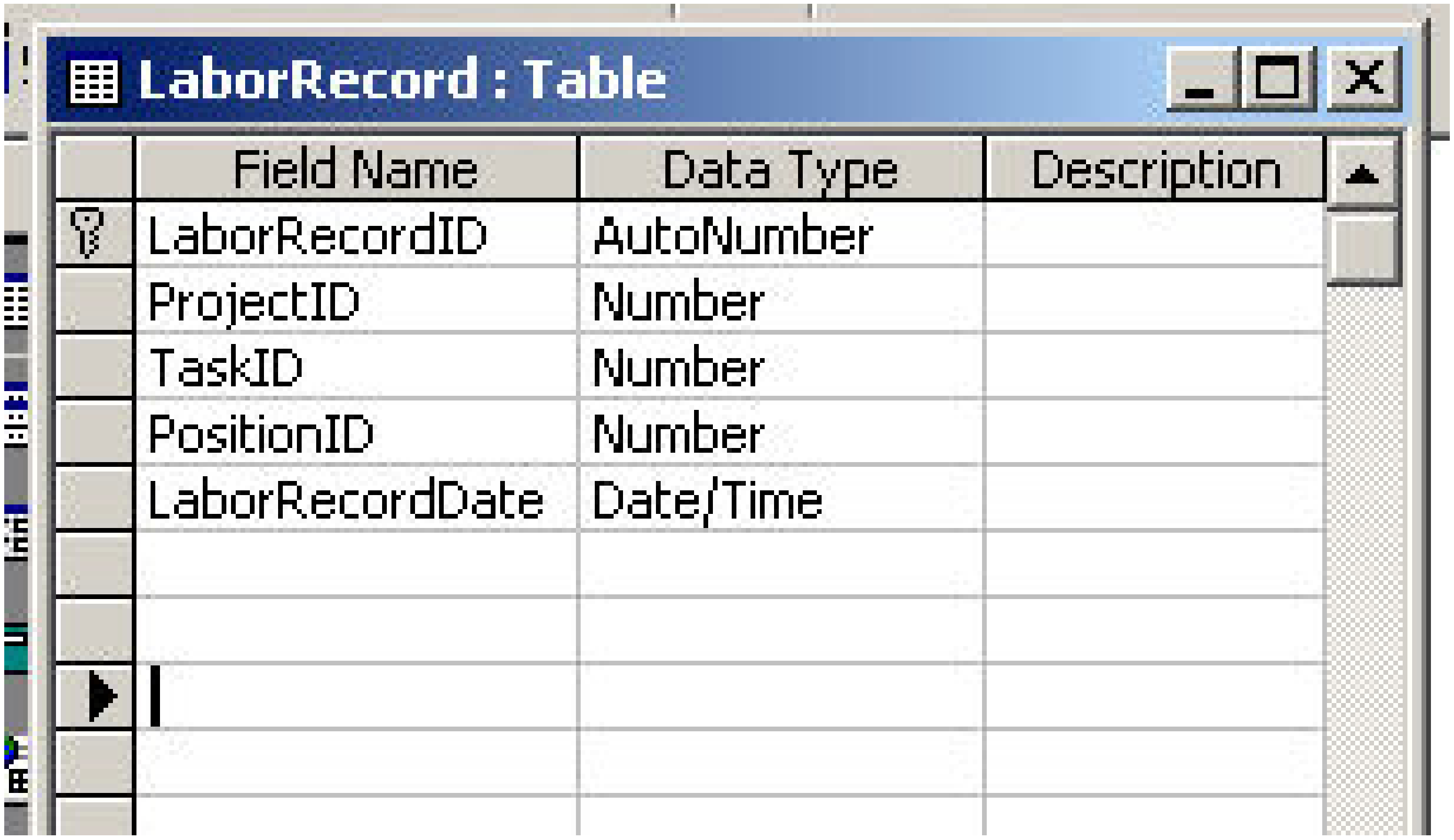
This virtual Time table does not have to be created for us to use its Primary Key (e.g.: Date) as a Foreign Key in another table as a Time Relation.

The next relation makes use of this virtual Time table.

1) A ProjectTaskPositionAssignment may be worked on (be related to) more than one Date.

2) On a specific Date, work may be done for (be related to) more than one ProjectTaskPositionAssignment.

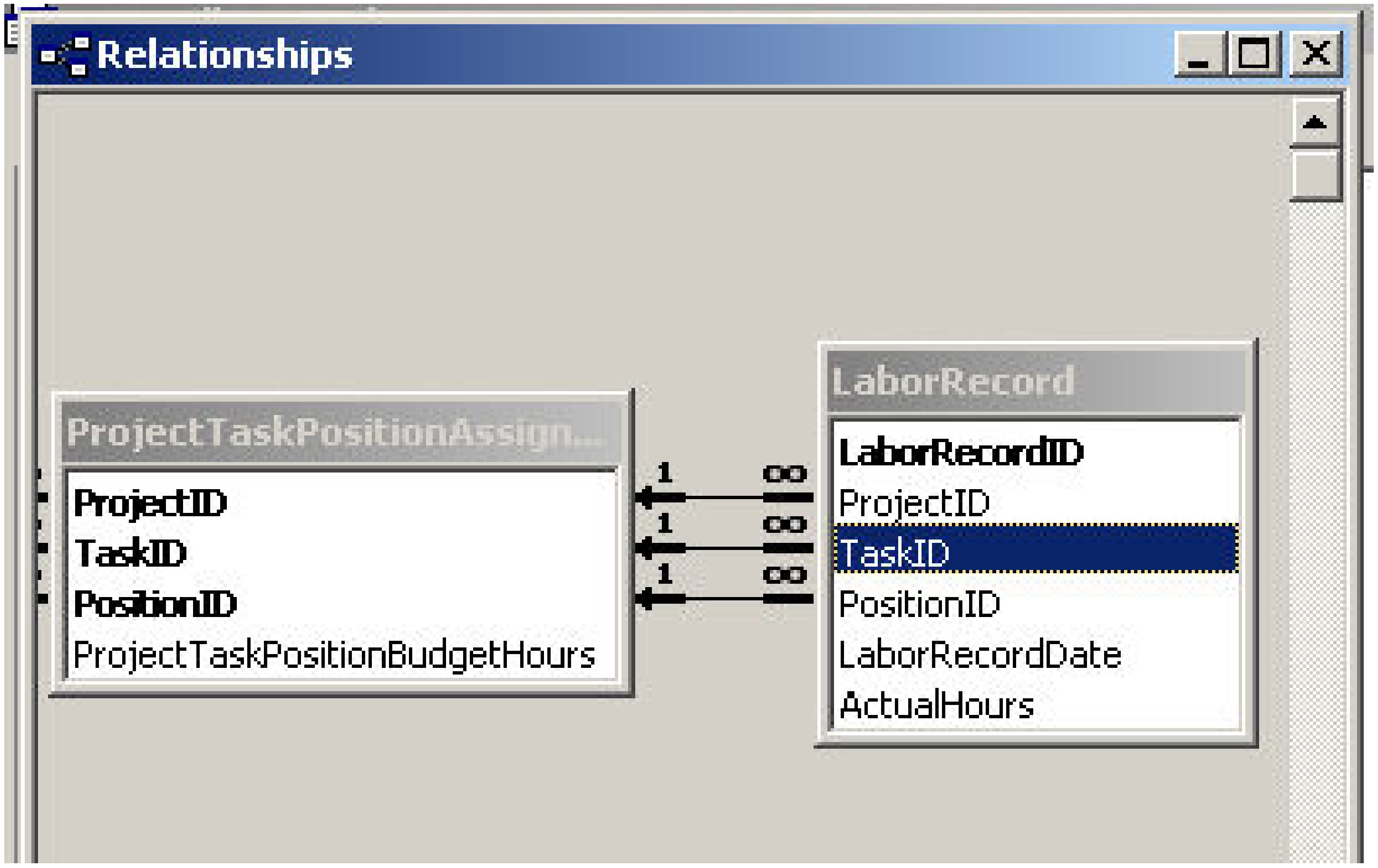
This identifies another many-to-many relation. To represent this, another **relation table** is created:



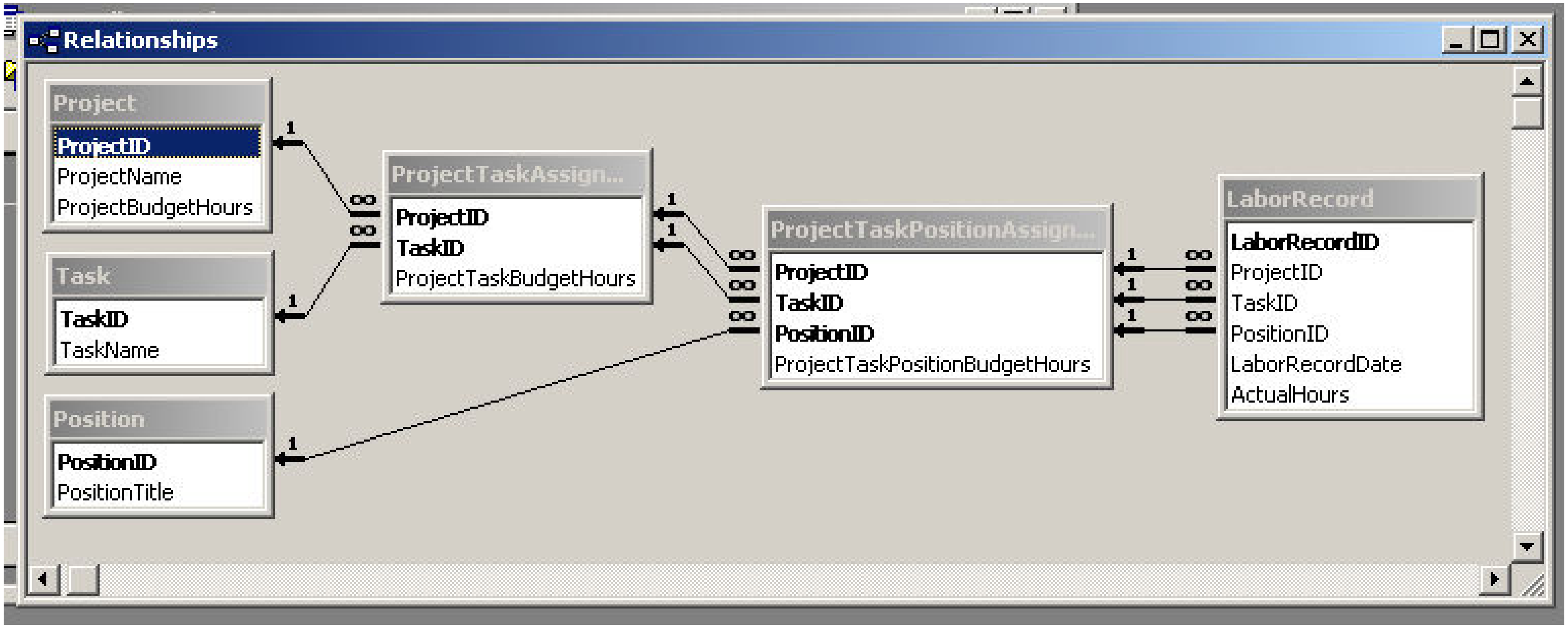
Field Name	Data Type	Description
LaborRecordID	AutoNumber	
ProjectID	Number	
TaskID	Number	
PositionID	Number	
LaborRecordDate	Date/Time	

LaborRecord table contains a record of all Dates having work done on the list of ProjectTaskPositionAssignments. The Primary Keys from ProjectTaskPositionAssignment and **virtual Time table** (Date) are brought into the LaborRecord table as **Foreign Keys** and could be combined to form its **Primary Key**.

A decision was made, however, to create a **system assigned number** to use as the **Primary Key**, instead.

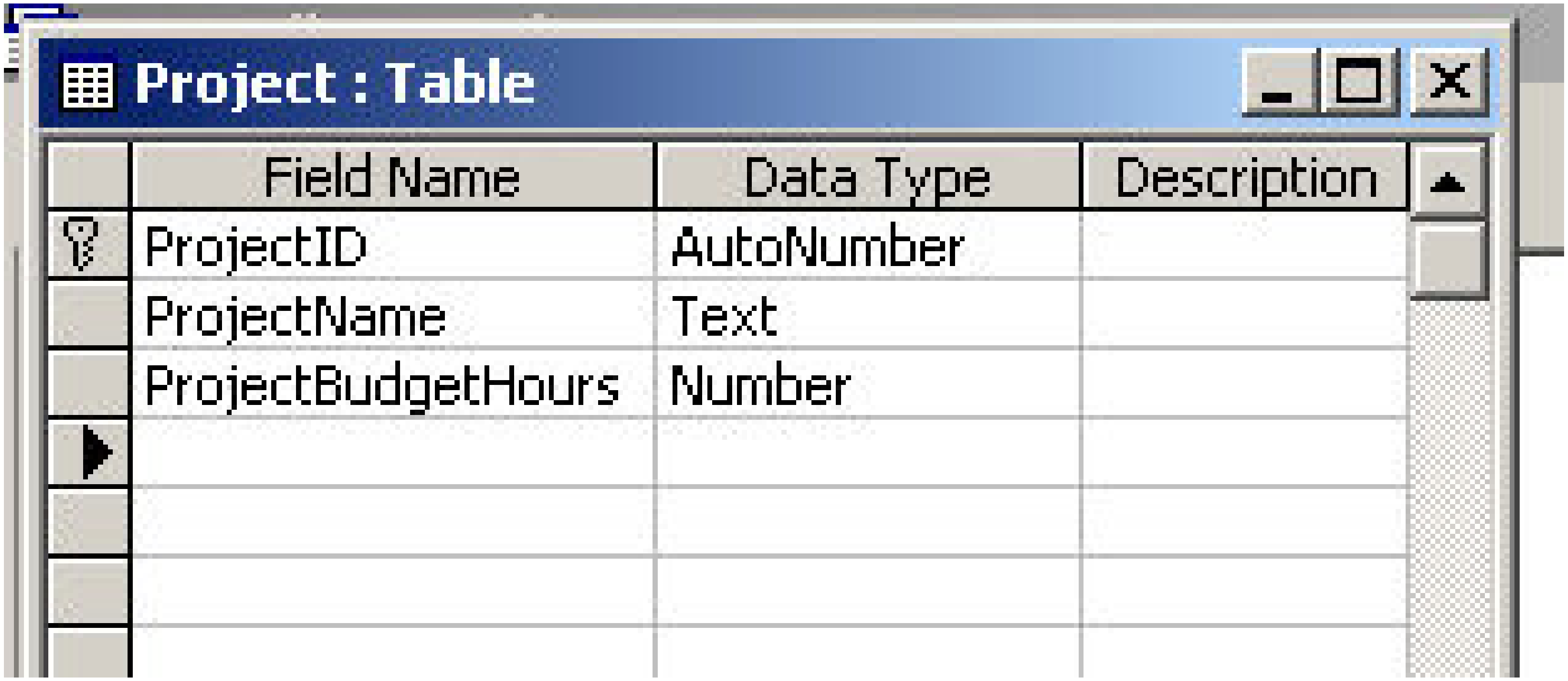


All of the important Entities and their Relations have been defined by a set of tables



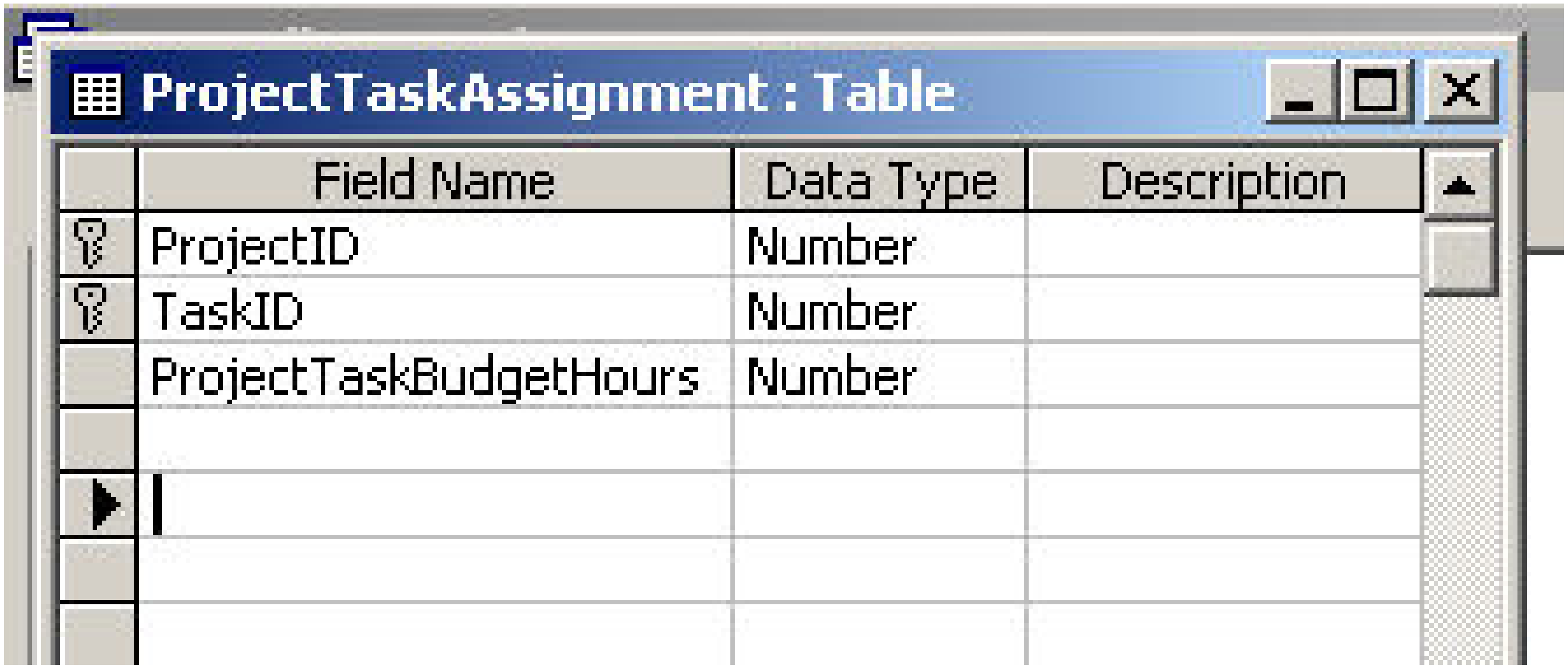
There are some other bits of information (Attributes) that must be put somewhere. This is done by, examining each of the Entity tables and Relation tables to find the most suitable home for these Attributes.

The total budgeted hours for a Project was added to the Project table as ProjectBudgetHours.



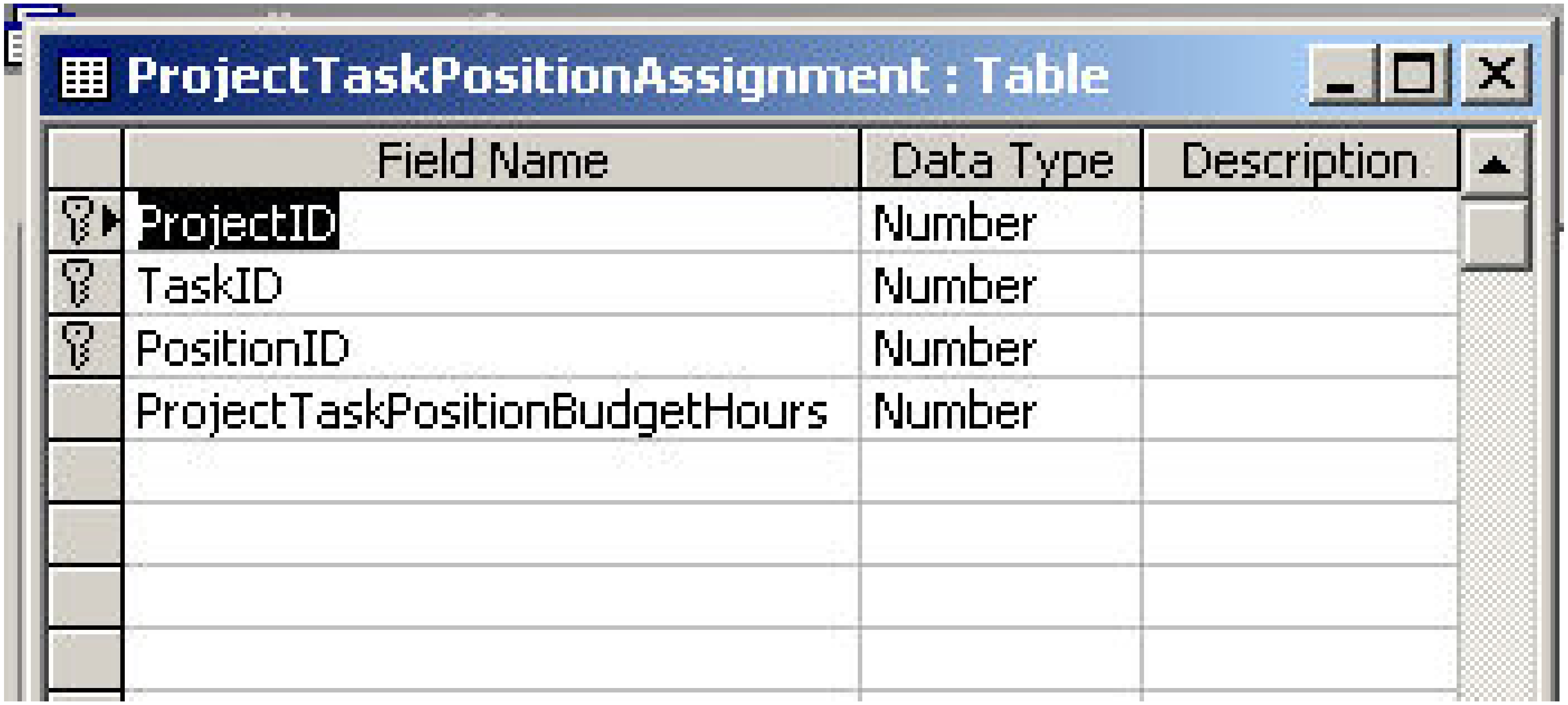
	Field Name	Data Type	Description
🔑	ProjectID	AutoNumber	
	ProjectName	Text	
	ProjectBudgetHours	Number	
▶			

These budgeted hours must be divided amongst the tasks assigned to a Project, so they were added to the ProjectTaskAssignment table as ProjectTaskBudgetHours.



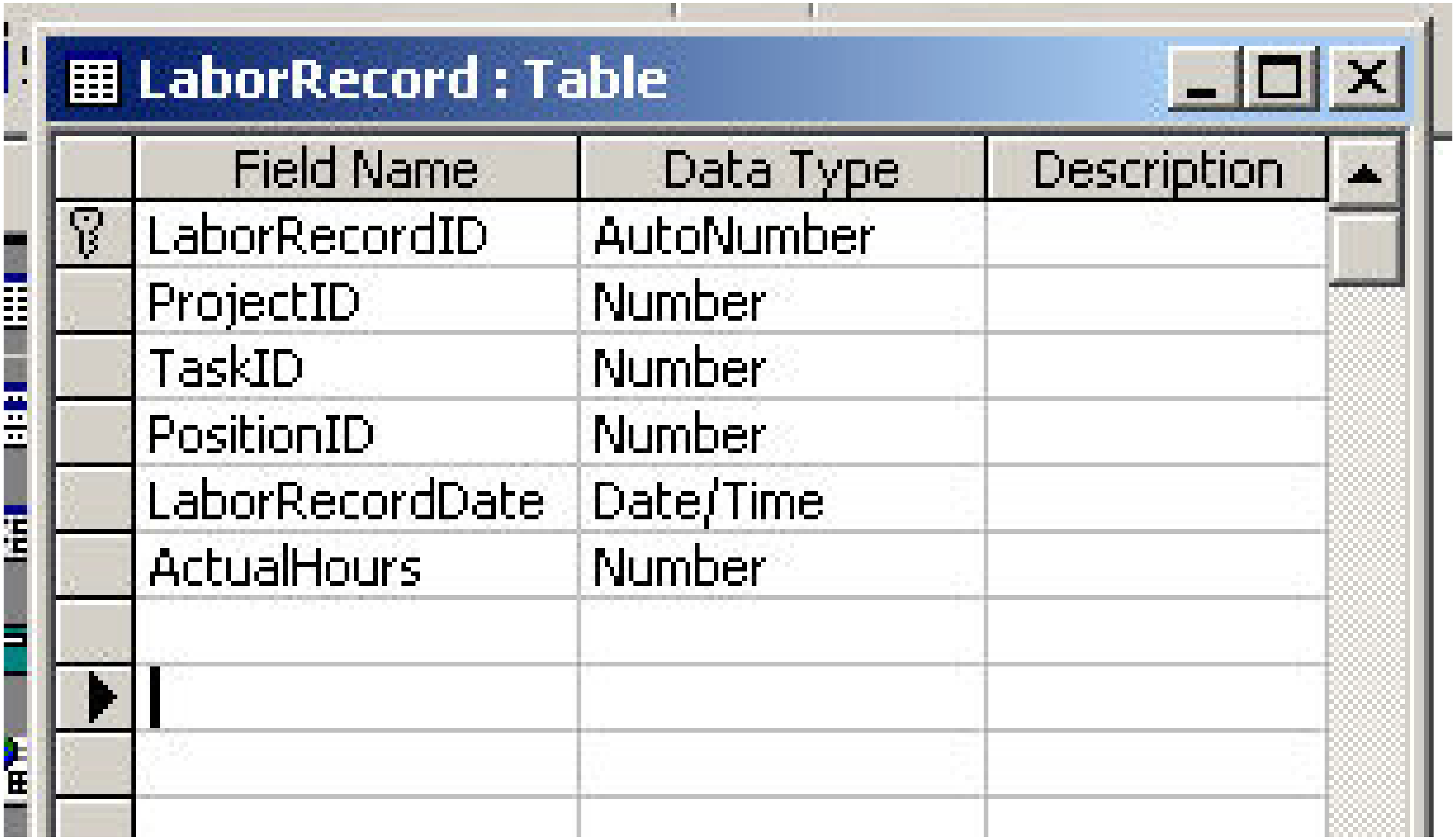
Field Name	Data Type	Description
ProjectID	Number	
TaskID	Number	
ProjectTaskBudgetHours	Number	

In a similar sense the budget hours assigned to a ProjectTask must be divided amongst the Positions assigned to that ProjectTask. This was added to the ProjectTaskPositionAssignment table as ProjectTaskPositionBudgetHours.


ProjectTaskPositionAssignment : Table

Field Name	Data Type	Description
ProjectID	Number	
TaskID	Number	
PositionID	Number	
ProjectTaskPositionBudgetHours	Number	

The ideal place to record the each day's hours of work by each Position on their assigned Project/Task was in the LaborRecord table as ActualHours.



LaborRecord : Table

Field Name	Data Type	Description
LaborRecordID	AutoNumber	
ProjectID	Number	
TaskID	Number	
PositionID	Number	
LaborRecordDate	Date/Time	
ActualHours	Number	

With this Data Model, every one was satisfied that hours of effort could be budgeted and tracked at the Project, Task and individual Position levels.

Can you think of somebody and some EntityTable that may have been left out?

How about Office Manager/Personnel-HR Manager?

I am sure they would have modified the Position Table and added a few tables for Name, HireDate, PayRate, PayRateEffectiveDate, …

Leaving them out does not help the Buy-In and overall success!