

1. Construção de uma Base de Dados em Access

Comecemos por conhecer um negócio que no caso poderá ser o da **inscrição de alunos em disciplinas**. Cada **aluno de um curso indicará por ordem as disciplinas** em que se pretende inscrever que será sujeita a uma **validação posterior** em função do número de alunos admissíveis.

O tema deve ser encarado como um subtema de um negócio bem mais complexo, pelo que se ressalvam todas as situações de ordem prática não abordadas.

De notar que na exposição do problema, identificamos de imediato qual a questão a resolver em vez de nos perdermos com descrições artísticas da situação...

Consideremos então que os **alunos estão matriculados num determinado curso** e que para esse **curso há uma colecção de disciplinas** nas quais os alunos se poderão inscrever.

Identificamos então as regras:

- O Aluno inscreve-se em disciplinas. "Inscreve-se" significa que tem de haver um suporte para esse facto, que terá características próprias, como sejam, por exemplo: a ordem de preferência, a data de ocorrência e a validação posterior.
- O Aluno está matriculado num curso
- Um Curso tem uma colecção de disciplinas nas quais o aluno se poderá inscrever.

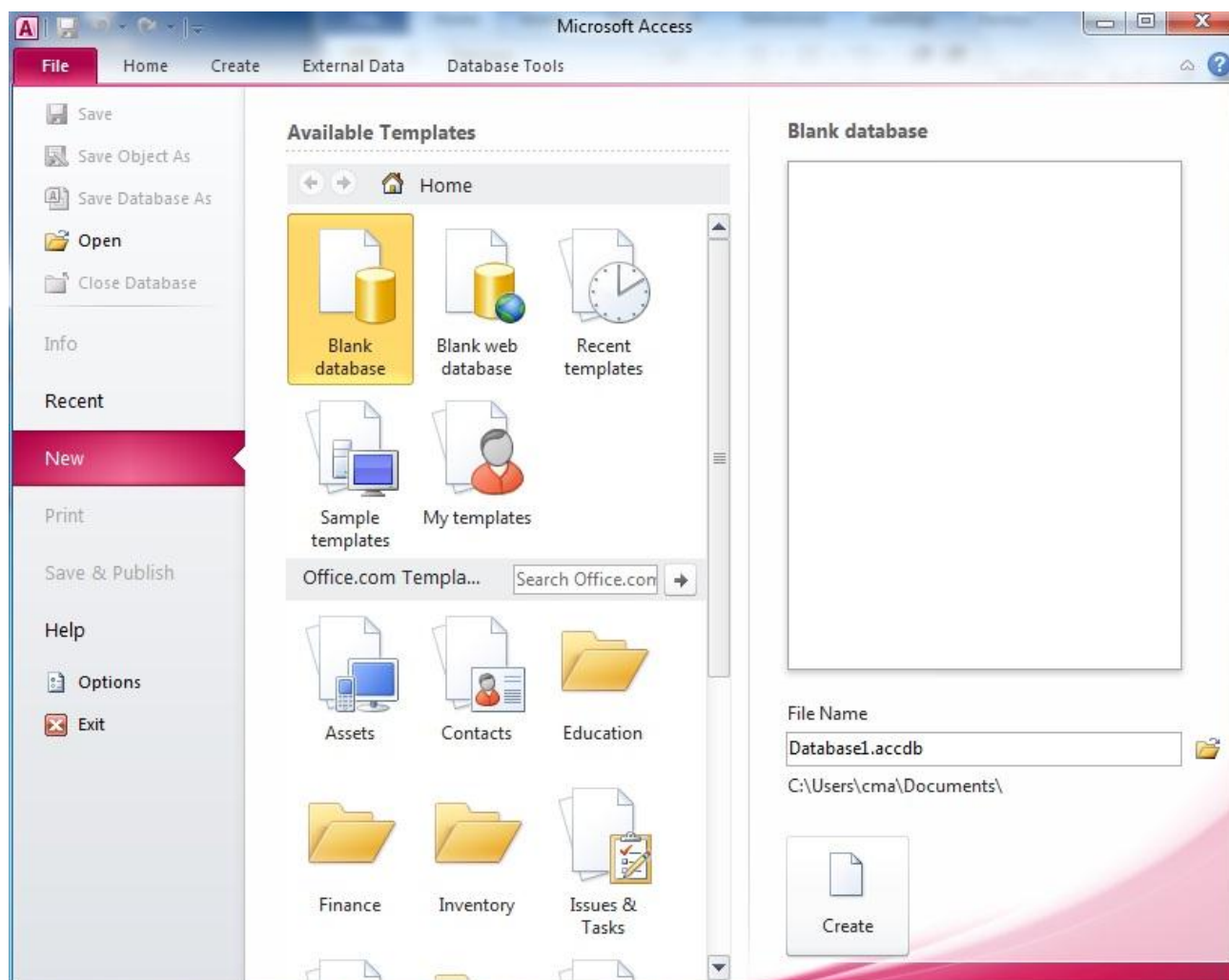
Que para serem suportadas precisam das seguintes entidades:

- ALUNO
- CURSO
- DISCIPLINA
- INSCRIÇÃO

Das regras de negócio e entidades consideremos como atributos os seguintes:

- ALUNO: Código interno, Nome, Telefone, Data de Nascimento e Curso em que está matriculado
- CURSO: Código e Designação
- DISCIPLINA: Código, Nome, Número máximo de alunos que se podem inscrever, Curso em que é leccionada, Ano, Unidades de Crédito
- INSCRIÇÃO: Aluno, Disciplina, Número de ordem e Data, Estado de validação

Estamos em condições de iniciar o Access e encontrar o quadro inicial:

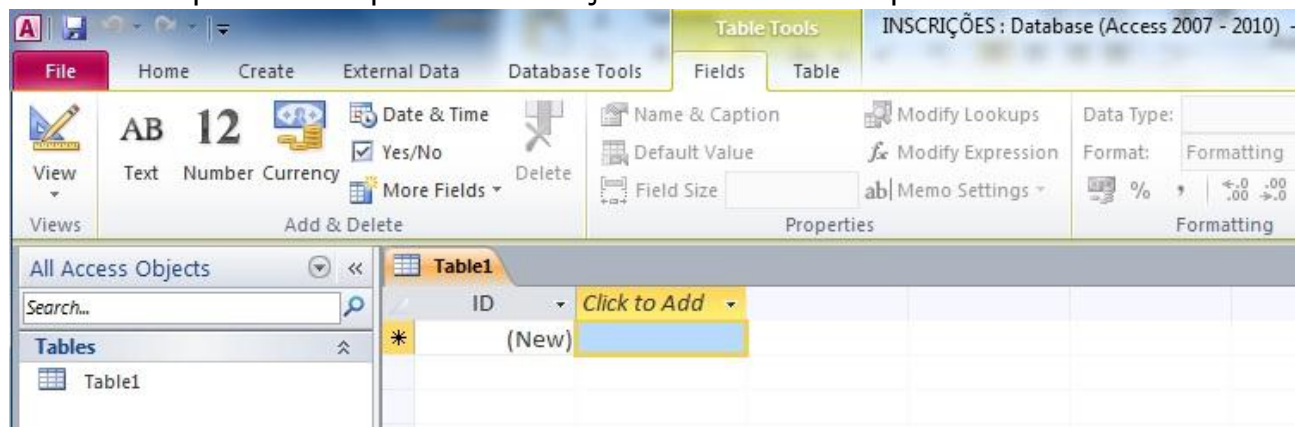


Como seria de esperar, haverá muitas formas de efectuar as operações necessárias. Uma das possíveis é mostrada neste documento.

Para criarmos uma nova base de dados bastará indicar o nome e efetuar um clique no botão [Create]. Previamente pode escolher o directório onde ficará localizado o ficheiro da base de dados bem como o ficheiro de controlo.

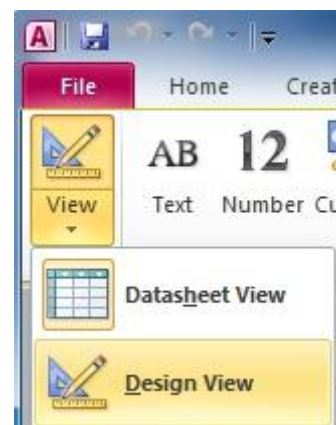
No nosso caso o nome poderá ser: INSCRIÇÕES e pressionando o botão [Create] aparece o quadro base de trabalho do Access.

Abre-se um quadro onde podemos começar a trabalhar numa primeira tabela:

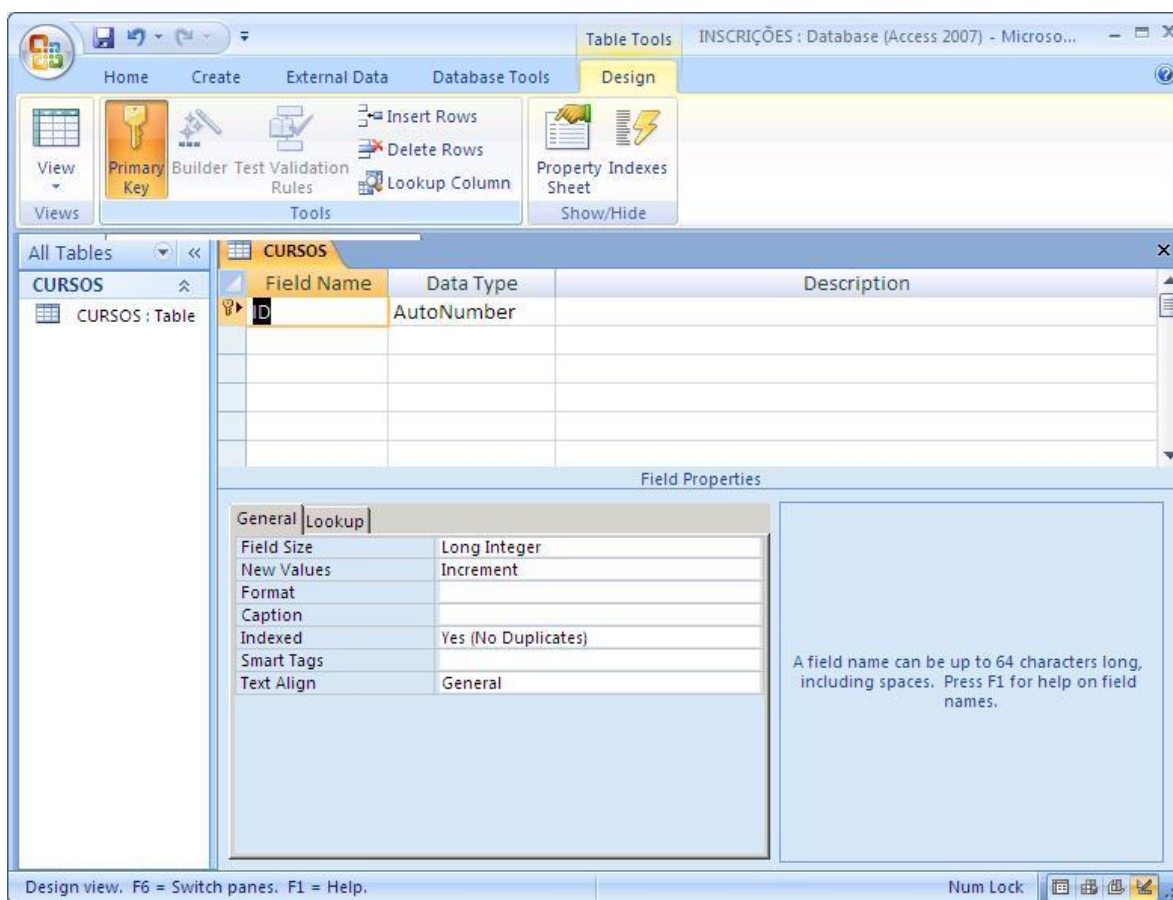


Mas vamos passar a um modo de vista de projecto de estrutura: "Design View":

Que nos vai obrigar a gravar a tabela com um nome. Como a tabela Cursos, não depende de mais nenhuma, comecemos por ela:



E passamos a ter uma vista onde poderemos então especificar os campos:

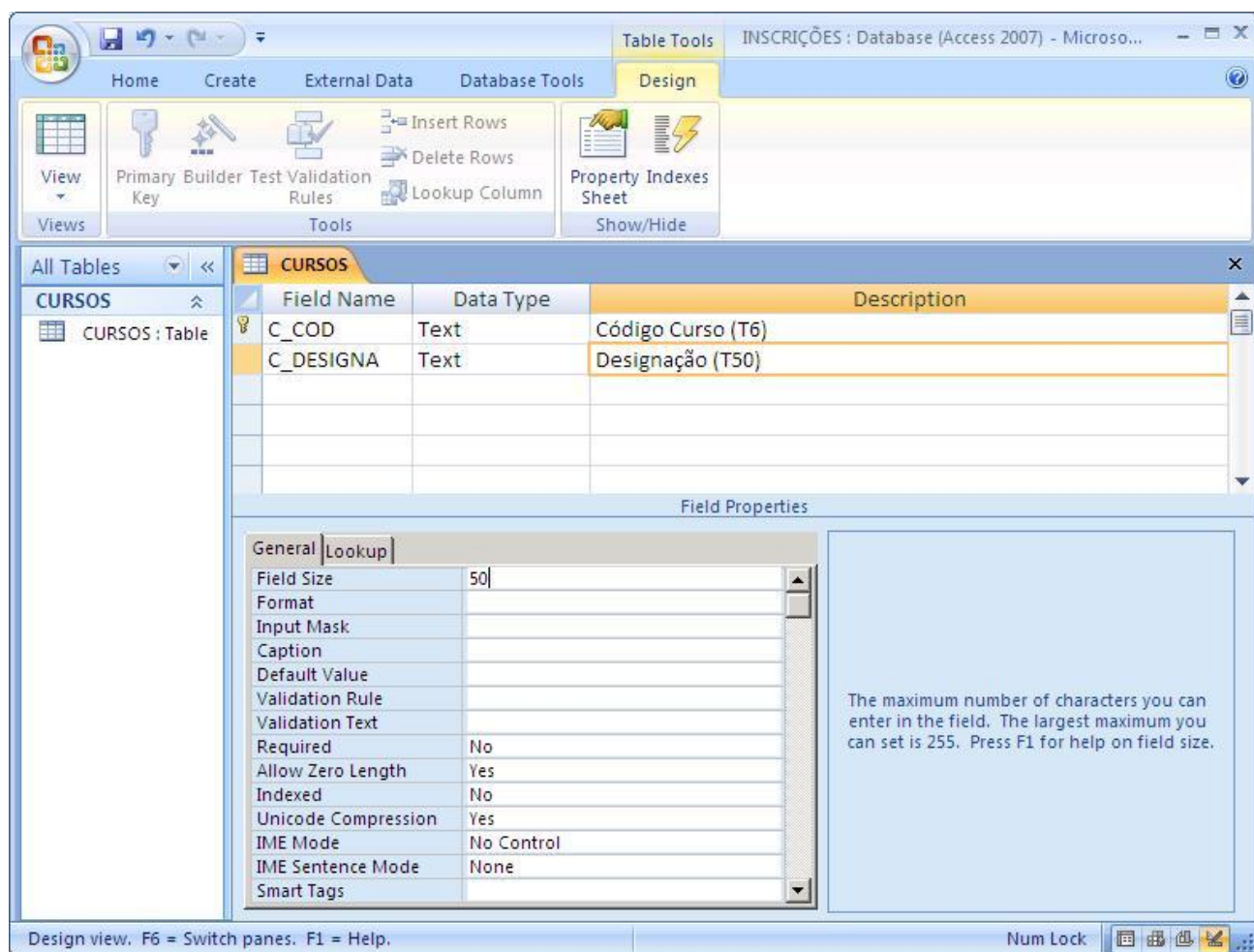


Para cada campo indicamos o seu nome, tipo de dados e escolhemos o tamanho e devemos sempre incluir uma descrição.

Para cada campo escolhemos um nome com um prefixo que tenha um ou dois (mais se necessário !!!) caracteres seguido de uma abreviatura da descrição do campo.

Note-se que na parte inferior da janela há uma zona para uma série de atributos de cada campo. Para melhor ilustrar o exemplo, indicamos o tipo e dimensão do campo na coluna da descrição.

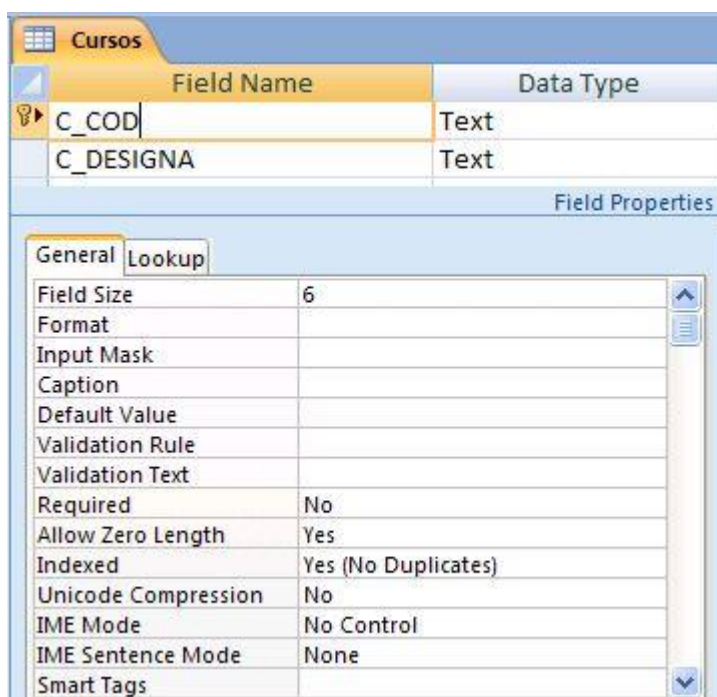
Ficariamos o seguinte aspecto:



O passo seguinte será escolher a chave primária para esta tabela e para isso posicionamos o cursor no campo escolhido: C_COD e vamos pressionar o botão que tem a chave:

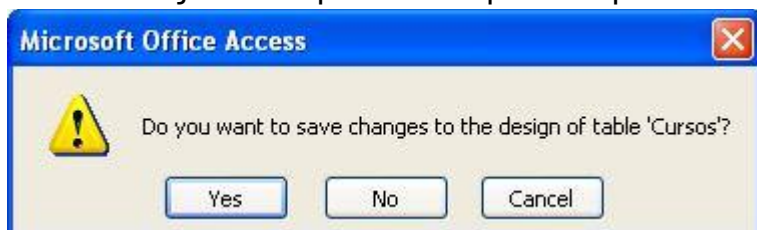


Note-se o aspecto com que fica a definição da tabela:

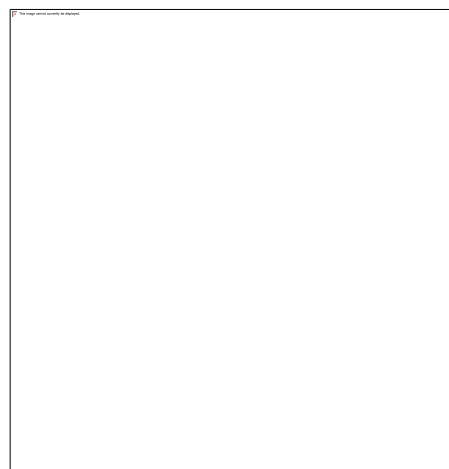


em particular a propriedade "Indexed" que passa a ter a opção "Yes (No Duplicates)" e o símbolo da chave antes do nome do campo.

A tabela está pronta, pelo que fazemos Save (File/ Save, Ctrl+S ou botão de gravação) ou fechamos a janela e aparece um quadro a pedir a confirmação:



Para criarmos uma nova tabela temos de ir ao menu "Create" e escolher "Table"



Vamos então repetir o processo para as outras tabelas.

DISCIPLINAS : Table

Field Name	Data Type	Description
D_COD	Text	Código da Disciplina (T4)
D_NOME	Text	Nome (T40)
D_MAXINSCR	Number	Número máximo de inscrições (integer)
D_ANO	Number	Ano do Curso (integer)
D_UC	Number	Unidades de crédito (integer)
D_C_COD	Text	Curso (T6)

Field Properties

General Lookup

Field Size: 50

Format:

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Allow Zero Length: Yes

Indexed: Yes (No Duplicates)

Unicode Compression: Yes

IME Mode: No Control

IME Sentence Mode: Phrase Predict

Smart Tags:

De notar o nome escolhido para o campo onde irá ser armazenado o curso: **D_C_COD**, composto pelo prefixo comum **D_** e pelo nome do campo chave primária da tabela onde a ligação irá ser estabelecida. Este campo é uma chave estrangeira.

Na tabela dos alunos usaremos o mesmo princípio.

ALUNOS : Table

Field Name	Data Type	Description
A_COD	Text	Código Aluno (T8)
A_NOME	Text	Nome (T60)
A_TEL	Text	Telefone (T12)
A_DN	Date/Time	Data nascimento (dd-mm-yyyy)
A_NUMDISC	Number	Número de disciplinas em que está inscrito (I)
A_C_COD	Text	Código Curso (T6)

Field Properties

General Lookup

Format: Short Date

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Indexed: No

IME Mode: No Control

IME Sentence Mode: Phrase Predict

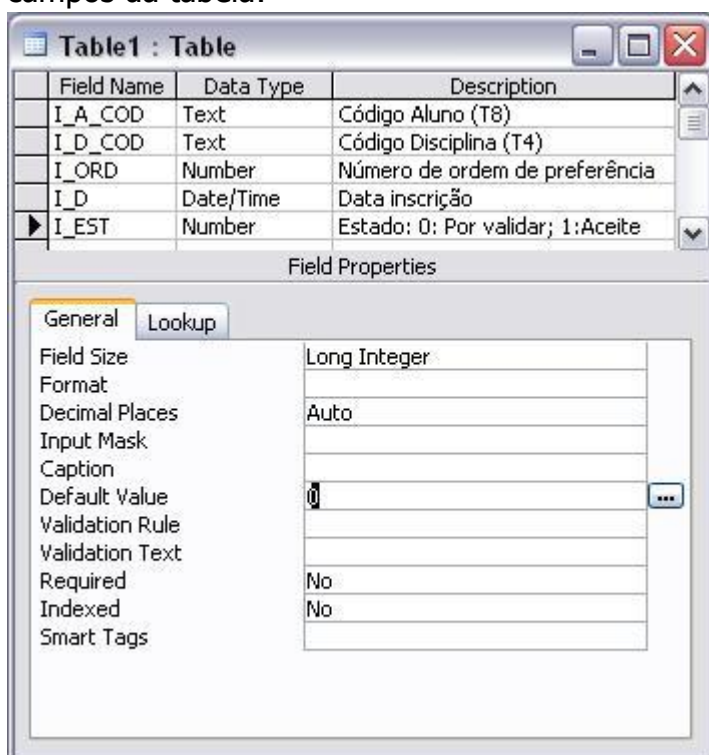
Smart Tags:

Finalmente a tabela para as inscrições. Esta tabela é a mais importante de todo o problema. Enquanto as outras são estáticas, podem ter mais ou menos registos, mas cada um é perfeitamente autónomo, nesta tabela ficará armazenada toda a dinâmica do negócio.

Esta tabela, que em muitos dos métodos de análise é designada por tabela associativa, e aparece quase como um consequência menor pelo facto de desdobrar a relação de N – M que ocorre entre a tabela dos Alunos e a das Disciplinas é que contém toda a informação importante do negócio. É neste tipo de tabelas que devemos focar toda a atenção e não devem surgir como consequência, mas sim como origem da necessidade das outras. De nada servirá ter uma tabela de Alunos, uma de Cursos e uma de Disciplinas se não se perceber que o negócio é gerir as inscrições. Temos é de perceber o que são as Inscrições e o que precisamos de fazer com elas e depois então encontrar as tabelas externas. No caso particular sabemos que as inscrições são efectuadas numa data, que são ordenadas em termos das preferências e que terão de ser validadas.

De uma forma mais genérica deveremos procurar encontrar as formas dinâmicas dos negócios e perceber a sua caracterização - podemos dizer que serão representadas pelas tabelas internas ou dinâmicas. Ao identificarmos o que elas associam encontramos as tabelas externas ou estáticas.

Nesta tabela haverá que ter cuidado com algumas questões. Começemos por criar os campos da tabela:



Field Name	Data Type	Description
I_A_COD	Text	Código Aluno (T8)
I_D_COD	Text	Código Disciplina (T4)
I_ORD	Number	Número de ordem de preferência
I_D	Date/Time	Data inscrição
I_EST	Number	Estado: 0: Por validar; 1: Aceite

Field Properties	
General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	No
Smart Tags	

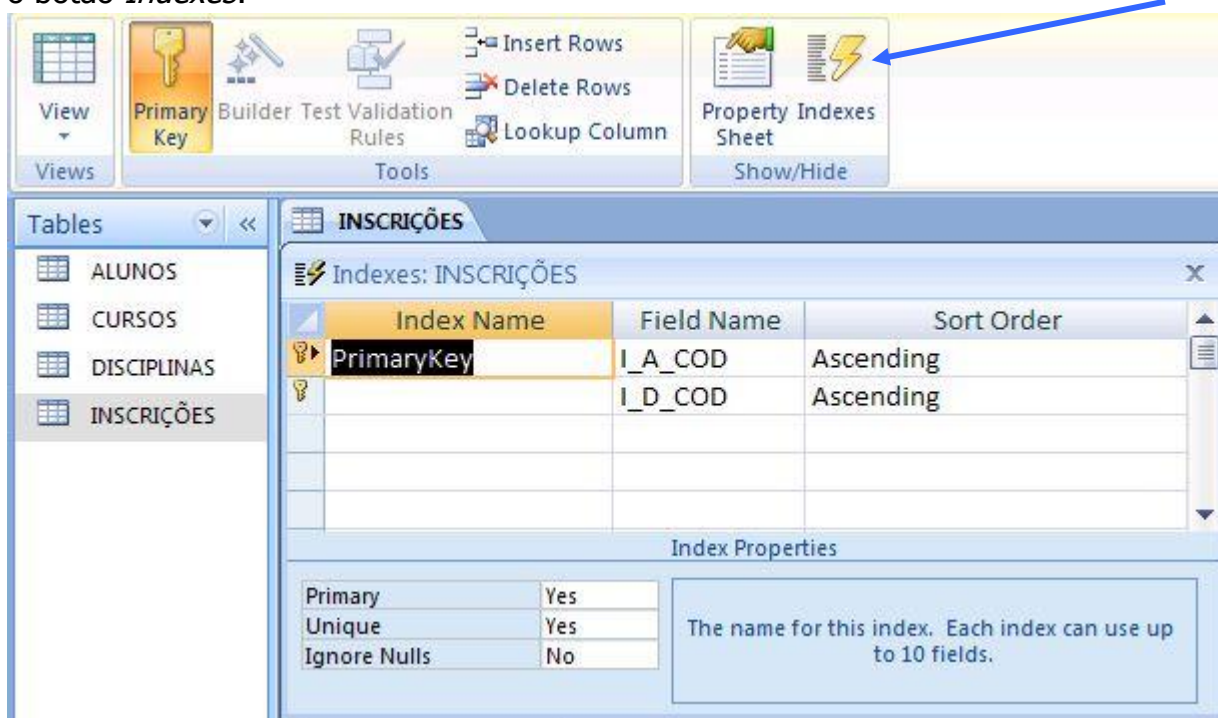
A escolha da chave primária terá de recair nos campos **I_A_COD** e **I_D_COD**, uma vez que cada aluno só irá inscrever-se na disciplina uma vez. Por outro lado o aluno deverá

indicar por ordem, a sua preferência nas inscrições, pelo que o par de campos **I_A_COD**, **I_ORD** também deverá ter valores únicos.

Vamos então seleccionar os dois campos para criação da chave primária levando o cursor até ao rectângulo antes de I_A_COD, pressionando o botão do rato e arrastando até à linha de baixo, seguidamente fazemos um clique no botão de chave primária. Os dois campos ficarão marcados como sendo a chave primária.

Field Name	Data Type
I_A_COD	Text
I_D_COD	Text
I_ORD	Number
I_D	Date/Time
I_EST	Number

Para conseguirmos a outra chave única teremos de abrir a janela dos índices, pressionando o botão *Indexes*:



The screenshot shows the 'Indexes' window for the 'INSCRIÇÕES' table. The 'Primary Key' index is defined with fields 'I_A_COD' and 'I_D_COD' in ascending order. The 'Index Properties' section shows 'Primary' as Yes, 'Unique' as Yes, and 'Ignore Nulls' as No.

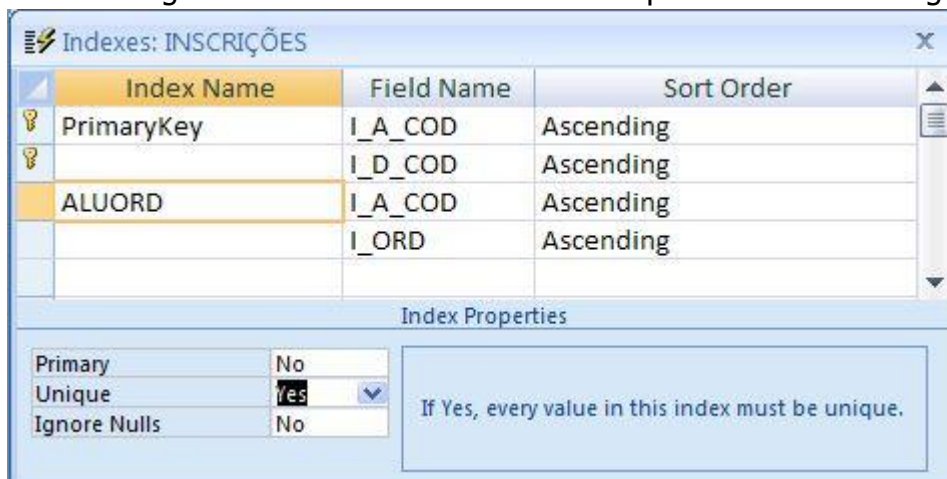
Index Name	Field Name	Sort Order
PrimaryKey	I_A_COD	Ascending
	I_D_COD	Ascending

Index Properties	
Primary	Yes
Unique	Yes
Ignore Nulls	No

The name for this index. Each index can use up to 10 fields.

Na janela que se abre pode verificar-se que a chave primária está definida e tem os campos pretendidos. Na parte inferior da janela podem observar-se as suas propriedades: Primária e Única.

Para conseguirmos outra chave única vamos preencher com a seguinte informação:



Index Name	Field Name	Sort Order
PrimaryKey	I_A_COD	Ascending
	I_D_COD	Ascending
ALUORD	I_A_COD	Ascending
	I_ORD	Ascending

Index Properties

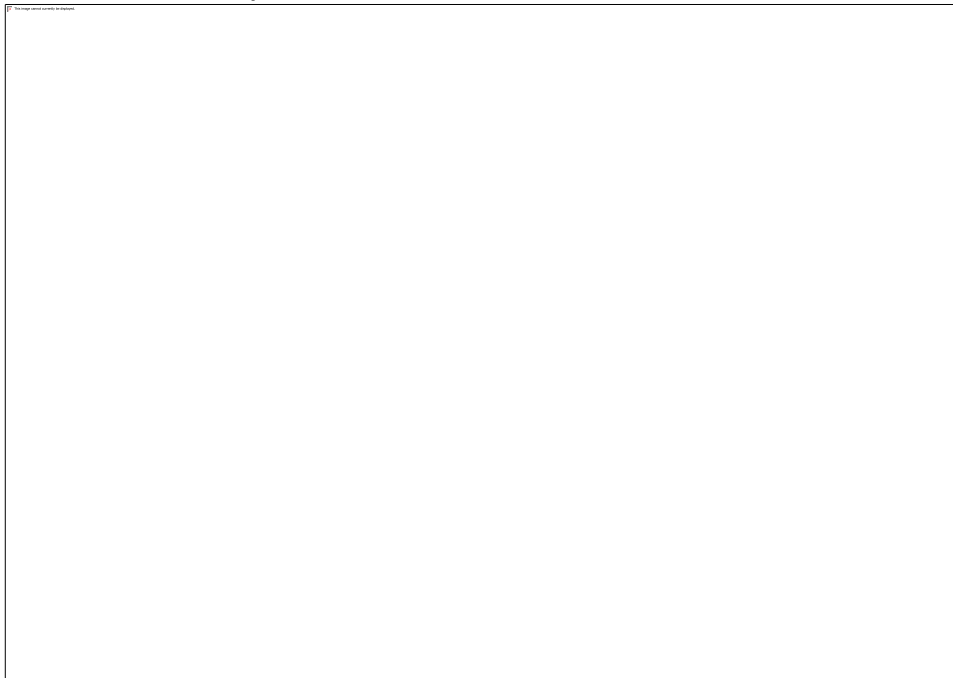
Primary:
 Unique:
 Ignore Nulls:

If Yes, every value in this index must be unique.

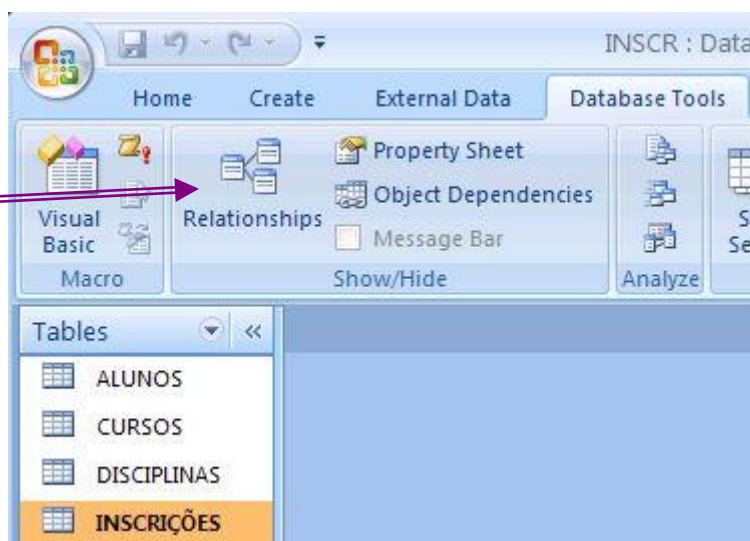
Deste modo iremos garantir que um aluno não pode colocar duas disciplinas com a mesma ordem de preferência.

Depois deste passo, fechamos a janela dos índices e gravamos a tabela. Podemos ter de gravar todas as tabelas, dependendo do modo como as fomos manipulando nesta fase do projecto.

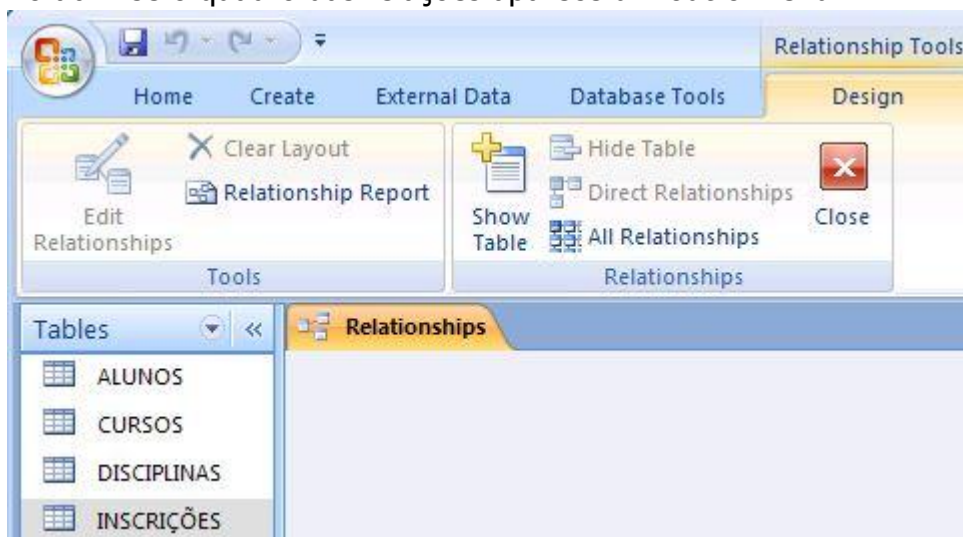
Ficamos com as quatro tabelas criadas:



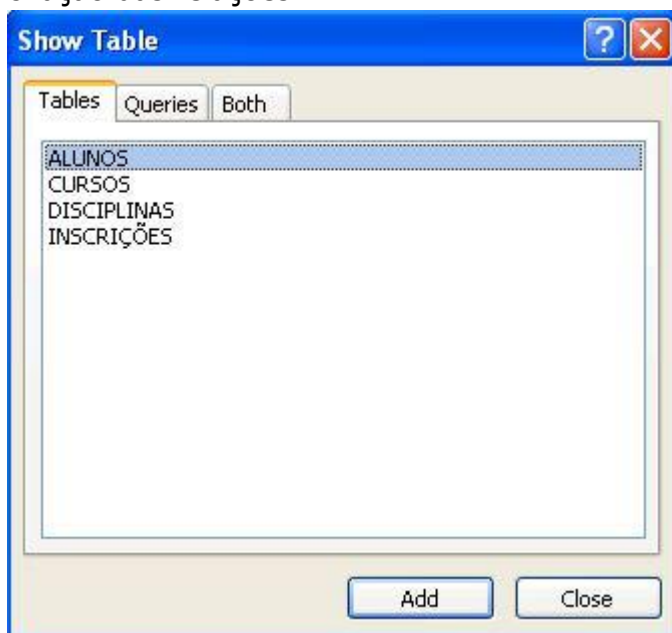
O passo seguinte será estabelecer as relações entre as tabelas. Para isso escolhe-se o menu "Database Tools" e faz-se um clique em: "Relationships"



Ao abrir-se o quadro das relações aparece um outro menu:



Com um clique em Show Table abre-se uma janela para selecção das tabelas para a criação das relações:



Como pretendemos um diagrama global podemos seleccionar as tabelas todas e dar-lhes a seguinte disposição:



Para criarmos as relações entre as tabelas, iremos com o cursor fazer um clique numa chave primária e por arrasto levá-la até ao campo com o qual pretendemos estabelecer a relação. Ao largarmos o cursor irá aparecer um quadro para confirmarmos a relação entre as tabelas, através dos campos em questão e simultaneamente poderemos especificar que pretendemos uma relação de integridade referencial:

Edit Relationships

Table/Query: CURSOS Related Table/Query: ALUNOS

C_COD A_C_COD

☒ Enforce Referential Integrity

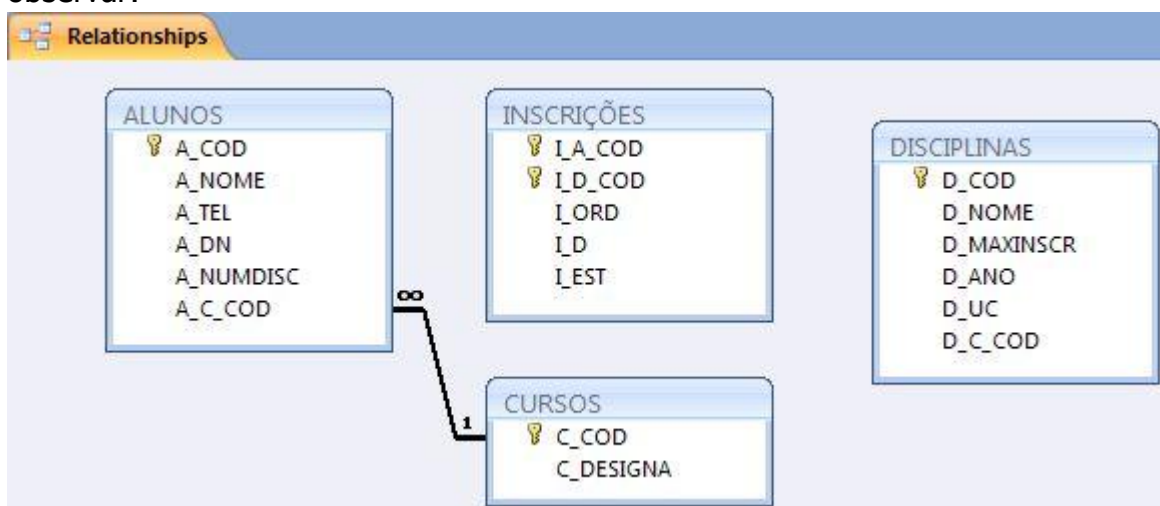
☐ Cascade Update Related Fields

☐ Cascade Delete Related Records

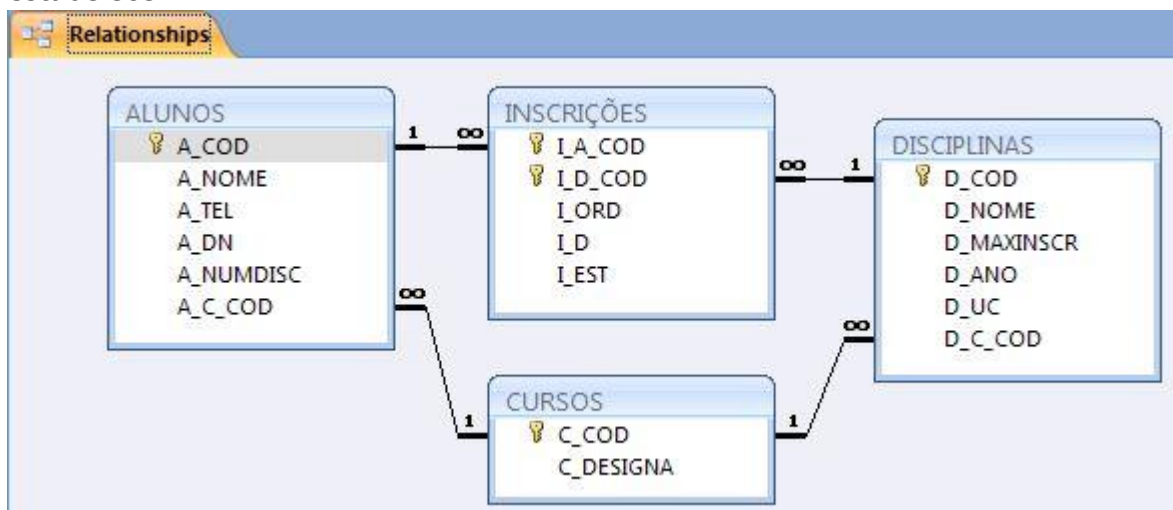
Relationship Type: One-To-Many

Buttons: Create, Cancel, Join Type..., Create New...

Pressionamos o botão [Create] e voltamos ao quadro das relações onde podemos observar:



Efectuamos a mesma operação para todas as outras relações que pretendemos estabelecer:



Fechamos a janela e temos a nossa base de dados pronta a ser usada.
O passo seguinte será criar uma interface, por exemplo através do VB.NET ou eventualmente através do próprio Access.

Uma vez que o motor do Access não dispõe de mecanismos internos de programação (nomeadamente de Triggers) a maior parte das regras de negócio terão de ser implementadas com a interface. Preferencialmente deveriam ser asseguradas por mecanismos internos ao motor da base de dados.

A título de curiosidade apresentamos as instruções SQL que permitem criar o mesmo modelo de dados no DBMS SQL Server.

```
CREATE TABLE CURSOS (
    C_COD          nvarchar (6)    NOT NULL ,
    C_DESIGNA      nvarchar (50)   NULL )
```

```
CREATE TABLE INSCRIÇÕES (
    I_A_COD        nvarchar (8)    NOT NULL ,
    I_D_COD        nvarchar (4)    NOT NULL ,
    I_ORD          smallint        NULL ,
    I_D            datetime        NULL ,
    I_EST          int             NULL )
```

```
CREATE TABLE ALUNOS (
    A_COD          nvarchar (8)    NOT NULL ,
    A_NOME         nvarchar (60)   NULL ,
    A_TEL          nvarchar (12)   NULL ,
    A_DN           datetime        NULL ,
    A_NUMDISC      smallint        NULL ,
    A_C_COD        nvarchar (6)    NULL )
```

```
CREATE TABLE DISCIPLINAS (
    D_COD          nvarchar (4)    NOT NULL ,
    D_NOME         nvarchar (40)   NULL ,
    D_MAXINSCR     smallint        NULL ,
    D_ANO          smallint        NULL ,
    D_UC           smallint        NULL ,
    D_C_COD        nvarchar (6)    NULL )
```

```
ALTER TABLE CURSOS ADD
    CONSTRAINT PRIMARY KEY (C_COD)
```

```
ALTER TABLE INSCRIÇÕES ADD
    CONSTRAINT DEFAULT (0) FOR I_EST,
    CONSTRAINT PRIMARY KEY (I_A_COD, I_D_COD)
```

```

CREATE INDEX ON INSCRIÇÕES (I_A_COD)
CREATE INDEX ON INSCRIÇÕES (I_D_COD)
CREATE UNIQUE INDEX ALUORD ON INSCRIÇÕES (I_A_COD, I_ORD)

ALTER TABLE ALUNOS ADD
    CONSTRAINT PRIMARY KEY (A_COD)

CREATE INDEX ON ALUNOS (A_C_COD)

ALTER TABLE DISCIPLINAS ADD
    CONSTRAINT PRIMARY KEY (D_COD)

CREATE INDEX ON DISCIPLINAS (D_C_COD)

ALTER TABLE ALUNOS ADD
    CONSTRAINT FOREIGN KEY
    (A_C_COD) REFERENCES CURSOS (C_COD)

ALTER TABLE DISCIPLINAS ADD
    CONSTRAINT FOREIGN KEY
    (D_C_COD) REFERENCES CURSOS (C_COD)

ALTER TABLE INSCRIÇÕES ADD
    CONSTRAINT FOREIGN KEY
    (I_A_COD) REFERENCES ALUNOS (A_COD) ,
    CONSTRAINT FOREIGN KEY
    (I_D_COD) REFERENCES DISCIPLINAS (D_COD)

```

De notar a forma organizada como as instruções aparecem. Primeiro a criação das tabelas e seus campos com os atributos básicos (tipo e aceitação de valor nulo, ou não), depois para cada tabela a criação da chave primária, outros atributos (valor predefinido) e índices e finalmente as ligações de integridade referencial.