**1.1 Databases and Data Types**

Clinicians' everyday workflow has been implanted into pieces of software, like Cerner Millennium to make their lives easier. It stores all of the data they need to do their job. Computers store data in many ways, but one of them are in databases.

A **database** is a means of storing information in such a way that it can be retrieved easily. In its simplest form, a relational database presents information in tables that contains rows and columns. A **table** is a collection of data of the same type (**rows**). Columns (**fields**) represent discrete data elements that describe each row.

Millennium contains a table called PERSON that stores all of the people registered in the system. Each row within the PERSON table is a different person and is described by the fields on the PERSON table. Each person has a PERSON\_ID, SEX\_CD, BIRTH\_DT\_TM and many other fields that make up that individual person.

Machine generated alternative text:
Selected Tables 
PERSON 
Aias 
PERSON ID 
PERSON STATUS CD 
PERSON TYPE CO 
PURGE OPTION CD 
RACE CO 
RELIGION CD 
RESIDENT CD 
ROWD 
SEX AGE CHANGE IND 
SEX CO 
SPECIES CD 
9 
TXN ID TEXT 
UPDT APPLCTX 
UPDT CNT 
9 
UPDT DT TM 
UPDT ID 
UPDT TASK 
VET MILITARY STAT 
12 
DQE 

Fields are comprised of two elements - a name and a data type. The name of the field is how we reference it in a query and the data type is the type of data that field stores.

**Numbers**

|  |  |
| --- | --- |
| i2 | An integer (-32,768 to +32,767) |
| i4 | An integer (-2,147,483,648 to +2,147,483,647) |
| f8 | A double/float number with a decimal point (0.29x1038 to 1.7x1038) |

Computers have several storage formats for numbers that are integers (whole numbers) and float numbers (numbers with a decimal point).

Integers with an **i2** data type can store any whole number between -32,768 and +32,768, while integers with an **i4** data type can store numbers between -2,147,483,648 and +2,147,483,647.

The main thing to do with numbers is arithmetic. Operations such as addition or subtraction take two number values and produce a new number. When applying an operator on two integers, the result is **always an integer**.

5 + 48

// → 53

10 / 4

// → 2 diving an integer by an integer will always result in an integer

Double or float values are numbers containing a decimal point and are described as being an **f8** data type. Mathematic operations applied to double numbers are precise (to a certain degree) and therefore will produce a more accurate result.

10 / 4.0

// → 2.5 diving an integer by a double or visa-versa results in a double

**Strings**

|  |  |
| --- | --- |
| CN | A fixed-length character string, where N is the number of characters |
| VC | A variable-length character string with a maximum of 524,288 characters |

Strings are used to represent text. They can be written by enclosing their contents in quotes.

"I love CCL"

'CCL is a great language'

You can use single quotes or double quotes as long as the start and end of the string match. Almost anything can be put between quotes and CCL will understand it as a string. But a few characters are more difficult.

You can imagine how putting quotes between quotes might be hard. *Newlines* (the characters you get when you press ENTER), *carriage returns* (used to reset a device's position to the beginning of a line of text), tabs and several other characters work a bit differently. These characters are called *control* characters and will be discussed more in depth later in the book.

Each letter, punctuation mark, space or digit in a string is called a character. Strings within Cerner can be either a fixed-length or a variable-length string.

**Fixed-Length Strings**

Fixed-length strings have a fixed number of positions assigned to them and are of **cn** data type where 'n' is the number of fixed positions.

If we have a string with a c5 data type and assign it the value "CCL", it would look like this.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | C | L |  |  |

The remaining two positions of unused space, contain nothing.

If we have a string with a c5 data type and try to assign the value "I love CCL", it will truncate all of the excess characters and leave us with five characters.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I |  | L | O | V |

So, why would you ever want to use fixed-string?

The answer is when the length of a string stays constant or you know the maximum length of the string. Take an example where you want to store each state's abbreviation in a database (e.g. "CA", "MO", FL", etc). Each abbreviation will only ever be two characters long, so it would make sense to use a **c2** data type to store this data.

**Variable-Length Strings**

While fixed-length strings have a pre-defined number of positions, variable-length strings do not. In Cerner, variable-length strings have a data type of **vc**. If we use the same two examples as above except try to assign each phrase to a vc string, this is what will happen:

|  |  |  |
| --- | --- | --- |
| C | C | L |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I |  | L | O | V | E |  | C | C | L |

There is no extra space and nothing gets truncated. The string assumes the size of the number of characters in it.

Different considerations need to be made depending on the data type Cerner defined to store data in a field. All of those considerations will be discussed throughout the book.

**Date and Time**

|  |  |
| --- | --- |
| dq8 | Used to store date and time |

Date and times are stored in a special format in the database and must be converted into a human-readable format to understand. Cerner's data type for date is called **dq8**. Understanding how this storage and conversion process works is fairly involved and will be thoroughly discussed in the date/time chapter.

**NULL**

There is a special value, NULL, that is used to denote the absence of data. NULL is itself a value, but it carries no information. NULL is not a data type, but it is worth mentioning because it can be used in a number of different ways that will become apparent as you progress through the book.

There are several other data types used within Millennium, but they are far less common and thus you only need to worry yourself with them in special circumstances. Numbers, strings, and date/time are by far the most common data types and the ones you should become the most familiar with.

**Fields**

Fields are comprised of two elements - a name and a data type. Field names are unique within a table. Two tables can often use the same field name. For instance, the PERSON table and ORDERS table both contain the field PERSON\_ID (more on this later).

Cerner has a specific naming convention when it comes to fields. The type of data stored in a field can often be determined purely by the name of the field. Cerner has standard suffixes it adds to field names that store a particular type of data.

|  |  |
| --- | --- |
| Suffix | Description |
| \_cd | A field that stores a code value. All code values within Cerner are f8 data type. Code values will be covered extensively in the code value chapter. |
| \_ind | A field with an i2 data type that stores either a 0 or a 1 where 0 represents false and 1 represents true. |
| \_id | A system assigned f8 number used to uniquely identify a row on a table. |
| \_dt\_tm | Date and Time field. |
| \_key | Data stored within a \_key field only contains letters and numbers (no dashes, periods, etc.) and is in all caps. These fields are excellent for looking up text strings that would be difficult otherwise. A person's last name, "Smith-Roberts" would be stored as "SMITHROBERTS". Key fields are either vc or cn data type and often times have a corresponding field that does contain punctuation. |
| \_flag | A field that stores an i2 value where each value has a specific meaning. For instance, when you place an order in Cerner, it can be a stand-alone order, from a CarePlan, from an Order Set, or a TPN or compound med. There is a flag field on the ORDERS table that denotes the type of order. |

Because multiple tables can contain the same field, the naming syntax I will use throughout the book so you don't get confused is *TABLE.field*. Thus, to reference the PERSON\_ID on the PERSON table I would write *PERSON.person\_id*.

**Exercise 1.1**

In your non-production environment, open up the application *DiscernVisualDeveloper.exe*. The application will be discussed in greater depth in the next chapter. For now, using it will simply reinforce the material covered above.

Once in the application, click on *File* and then select *New (or just type Ctrl + N)*. Ensure Blank is selected as the *File Type* and click *OK (or just press ENTER)*.

Machine generated alternative text:
File Type 
Program 
Prompt Program 
Ad Hoc Query 
Layout Program 
Program Wizard 
Free Form Label 
Discern Expert Template 
Program Name 
File Name 
LogicaWPath Save Location: 
CCLUSERDIR: 
Node: 
(default) 
Llse Header 

Navigate to *Tools -> Query Builder (Ctrl + Shift + Q)*. From here, select on the MILLENNIUM folder and in the Table Filter field at the bottom type in PERSON. As you type, the tables listed in the middle column should change. Double click on the PERSON table to move it to the Selected Tables column.

Machine generated alternative text:
TABLES FIELDS QUALIFICATIONS 
CATEGORIES 
COMMON 
FAVORITES 
INLINE 
PERSON ABORH 
MILLENNIUM 
DICTIONARY 
RDBMS 
ALL 
APPLICATIONS 
DOMAINS 
PERSON ANTIGEN 
12.1 RECORD STRUCTURES 
PERSON COMBINE 
DATA NOT COLL HIST 
Customize 
GROUP BY 
SORT 
CONTROL OPTIONS 
VIEW QUERY 
Selected Tables 
PERSON 
Run Query 
New Inline Table 
PERSON 
PERSON ABORH RESULT 
PERSON ALIAS 
PERSON ALIAS HIST 
PERSON ANTIBODY 
PERSON 
BENEFIT R 
PERSON 
BENEFIT SCH 
PERSON CODE VALUE R 
PERSON CODE VALUE 
R HIS T 
PERSON COMBINE BATCH 
PERSON COMBINE DET 
PERSON 
DATA NOT COLL 
PERSON 
PERSON 
DISMISSAL 
PERSON 
DISMISSAL ACTN 
PERSON 
Table F'her 

Click on the FIELDS tab and a list of all of the fields for the PERSON table are shown.

Machine generated alternative text:
TABLES 
FIELDS 
QUALIFICATIONS 
Aias 
GROUP BY 
SORT 
CONTROL OPTIONS 
VIEW QUERY 
Selected Helds 
Add apression 
at apression 
Run Query 
DQE 
DQE 
DQE 
DQE 
DQE 
Selected Tables 
PERSON 
9 
ABS BIRTH DT TM 
ACTIVE IND 
ACTIVE STATUS 
ACTIVE STATUS 
ACTIVE STATUS 
AGE AT DEATH 
AGE AT DEATH PRE 
AGE AT DEATH 
UNIT 
ARCHIVE ENV ID 
ARCHIVE STATUS CD 
ARCHIVE STATUS 
AUTOPSY CO 
BEG EFFECTIVE DT TM 
BIRTH 
DT CD 
9 BIRTH 
DT TM 
BIRTH 
PREC FLAG 
BIRTH TZ 
CAUSE OF DEATH 
Select 
Select Al Helds -+ 
• Show Helds 
GIO ssary 
o 

Write down the data types for the following fields:

Name\_full\_formatted \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sex\_cd \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Birth\_dt\_tm \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last\_first\_key \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Person\_id \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Active\_ind \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Spend the next couple of minutes familiarizing yourself with the first two tabs of this tool. Return to the TABLES tab and take a look a couple more tables (try *right clicking* on a field name or table name if you don’t know what it is!)

**1.2 Keys and Indexes**

**Keys**

We've discussed what a database is, but we really haven't covered how data is actually stored and retrieved. If we have a row on the PERSON table corresponding to an individual person, we need a way to uniquely access the row. What if we used a patient's name, birth date and SSN? Well, we don't always know this information for every patient and using three fields to uniquely identify a patient isn't very efficient.

Therefore, each row on a table has a unique identifier called a **primary key.** The primary key on the PERSON table is PERSON\_ID. Whenever a new person is added to the PERSON table, PERSON\_ID is incremented and assigned to that row. Therefore, we will never have two people with the same PERSON\_ID.

<insert image here>

If you take a look in Query Builder, there is a little red key next to PERSON\_ID. A red key indicates that this field is the primary key for the table.

Machine generated alternative text:
TABLES FIELDS QUALIFICATIONS 
Selected Tables 
PERSON 
GROUP BY 
SORT 
CONTROL OPTIONS 
Type 
VIEW QUERY 
Selected Helds 
PERSON 
PERSON STATUS CD 

If you right click on PERSON\_ID and view the properties, it conveniently tells us this is the primary key and it's a unique, system generated number.

Machine generated alternative text:
Table Name 
PERSON 
Sequence Name 
Person ID 
Field Properties 
Held Name 
PERSON ID 
PERSON ONLY SEQ 
This is the value of the unique primary dentifier of the person table It is 
an intemal system assigned number 

As mentioned before, different tables can have the same field name. This is very advantageous because it allows us to combine data from two different tables. Millennium is a **relational database**. A relational database is a database that is structured with relationships amongst the data stored within it. This is best explained with an example.

A patient comes into the ER and gets registered (a row is written to the PERSON table). The physician places several lab orders and eventually prescribes some medication and discharges the patient.

Orders within Millennium are stored on the ORDERS table.

**<insert image for orders and person>**

The orders table contains a field called the ORDER\_ID, many other fields describing the order, and a PERSON\_ID field! PERSON\_ID stores who the order is for. Since the PERSON\_ID is on the ORDERS table, but it is not the unique identifier for the order, it is called a **foreign key**.

The PERSON\_ID is the primary key on the PERSON table, but it's a foreign key on the ORDERS table. You can see this in Query Builder because there is now a yellow key next to PERSON\_ID instead of a red key.

Machine generated alternative text:
TABLES FIELDS QUALIFICATIONS 
Selected Tables 
ORDERS 
GROUP BY 
SORT 
CONTROL OPTIONS 
Type 
VIEW QUERY 
Selected Helds 
PERSON 

There is a relationship between PERSON table and the ORDERS table and it's through the PERSON\_ID. Because each order on the ORDERS table is stored with the person it's for, we can easily find all of the orders for any person.

We will be discussing in much greater depth how tables are connected and how we can join them together to connect disparate data in the database in Chapter 3.

For now, just make sure you understand the concept of primary and foreign keys.

**Indexes**

The topic of indexes could span several chapters. It's an extremely complex topic that involves an extensive knowledge of how computers store data to fully understand. The good thing is you don't need to fully understand it!

The only thing you need to know about indexes is that they are vehicles to improve lookup efficiency. The data within an index is stored in a different way than a row within a database and this allows the lookup of patient information to increase greatly.

**<show picture of data stored in database vs binary tree>**

The Discern team who developed DVDev has done us all a service and listed which fields within a table are indexes. Any field with any color key next to it is an index. This means we can search on this field to greatly improve efficiency.

Machine generated alternative text:
TABLES 
NAME FIRST 
NAME FIRST 
NAME LAST 
FIELDS 
QUALIFICATIONS 
GROUP BY 
SORT 
CONTROL OPTIONS 
VIEW QUERY 
Selected Helds 
vuoo 
VC202 
VC100 
Selected Tables 
PERSON 
9 
9 
NAME FIRST KEY 
NAME FIRST KEY A 
NAME FIRST KEY NLS 
NAME FIRST SYNONY 
NAME FULL FORMAT 
NAME LAST KEY 

The importance of indexes will be discussed in Chapter 3.

**Exercise 1.2**

In your non-production environment, open up the application *DiscernVisualDeveloper.exe*.

Once in the application, click on *File* and then select *New (or just type Ctrl + N)*. Ensure Blank is selected as the *File Type* and click *OK (or just press ENTER)*.

Machine generated alternative text:
File Type 
Program 
Prompt Program 
Ad Hoc Query 
Layout Program 
Program Wizard 
Free Form Label 
Discern Expert Template 
Program Name 
File Name 
LogicaWPath Save Location: 
CCLUSERDIR: 
Node: 
(default) 
Llse Header 

Navigate to *Tools -> Query Builder (or just type Ctrl + Shift + Q)*. From here, select on the MILLENNIUM folder and in the Table Filter field at the bottom type in ENCOUNTER. As you type, the tables listed in the middle column should change. Double click on the ENCOUNTER table to move it to the Selected Tables column.

Machine generated alternative text:
TABLES FIELDS QUALIFICATIONS 
CATEGORIES 
COMMON 
FAVORITES 
INLINE 
MILLENNIUM 
DICTIONARY 
RDBMS 
ALL 
APPLICATIONS 
DOMAINS 
12.1 RECORD STRUCTURES 
Customize 
GROUP BY 
SORT 
CONTROL OPTIONS 
VIEW QUERY 
Selected Tables 
Run Query 
New Inline Table 
ENCOUNTER 
ENCOUNTER 08282018 
Table F'her 
ENCOUNTER 

Click on the FIELDS tab and a list of all of the fields for the ENCOUNTER table are shown.

Machine generated alternative text:
TABLES FIELDS QUALIFICATIONS 
Selected Tables 
ENCOUNTER 
• Show Helds 
GIO ssary 
o 
GROUP BY 
Discern Query Builder 
SORT CONTROL OPTIONS VIEW QUERY 
Selected Helds 
ABN STATUS CD 
ACCIDENT RELATED I 
ACCOMMODATION 
ACCOMMODATION 
ACCOMMODATION 
ACCOMP BY CD 
ACTIVE IND 
ACTIVE STATUS CD 
ACTIVE STATUS DT 
ACTIVE STATUS pas 
ADMIT DECISION DT 
ADMIT EARLY IND 
ADMIT MODE CD 
ADMIT SRC CD 
ADMIT TYPE CO 
ADMIT WITH MEDICA 
ALC DECOMP DT TM 
Select 
Select Al fields -+ 
DQE 
DQE 
DQE 
Add apression 
at apression 
Run Query 

What is the primary key for the ENCOUNTER table?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Is the *loc\_building\_cd* field an index?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What kind of key is the *person\_id?*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What kind of data does the ENCNTR\_ALIAS table store?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is its primary key for the ENCNTR\_ALIAS table?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is the foreign key on the ENCNTR\_ALIAS table that connects ENCNTR to ENCNTR\_ALIAS?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Right click on the ENCNTR\_TYPE\_CD field and go to the properties. Notice that this field is a \_cd field while the other fields we viewed properties for were \_id fields.

Machine generated alternative text:
Table Namæ 
ENCOUNTER 
Description 
Encounter Type Code Value 
Definition 
Held Name 
ENCNTR TYPE CD 
Categorizes the encounter into a logical group or type aamples may 
inSIude inpatient. outpatient. etc 

Click on *Lookup* to see a list of all of the ENCNTR\_TYPE\_CDs in your environment.

We will be discussing code values and how they work in Chapter 4.

**Chapter 1 Recap**

* Millennium stores all of its data in a database. The Millennium database is made up of thousands of tables - each table contains different fields that describe the data on that table.
* A table stores many rows of data, like the PERSON table storing an individual row of data for each person.
* The most common data types used in Millennium are numbers (i2, i4, f8), strings (cn, vc), and date/time (dq8).
* Cerner has a specific naming convention for fields that have a specific purpose. The type of data stored in a field can often be determined by the name of the field.
* Each row on a table has a unique identifier called a primary key.
* The primary key on the PERSON table is the PERSON\_ID, but it is the foreign key on the ORDERS table.
* Right clicking on a field name or a table name in the Query Builder application allows you to view more information about that field or table.
* Indexes are ways to store information efficiently for lookup.