Introduction to databases and database management systems[1]

Contents

Database, table, column, row, entity

A **database is an organized collection of related information stored in a file.** A database for a **relational database management system** such as Oracle, SQL Server, MySQL, and Microsoft Access (mentioned because of their courses at Santa Monica College) has one or more tables, each with one or more columns and rows. A **column** is a synonym for a **field** or **attribute**. A **row** is a synonym for a **record**. A field also can be the value stored at the intersection of a row and column. An **entity** is a table or a group of tables.

Here's a table in Oracle format:

REPRESENTATIVE table

REPRESENTATIVE_ID	LAST_NAME	FIRST_NAME	REGION	HIRE_DATE	PHONE
11	Rogler	Harold	SW	05-JAN-99	(310) 456-7890
22	Higgins	Heather	SE	16-DEC-91	(404) 524-8472
33	Sullivan	Pat	NE	21-FEB-88	(305) 734-2987
44	Speed	Kristen	MW	14-JUN-90	(708) 823-8222
55	Sigafoos	Alex	NW	05-MAR-01	(310) 123-7890

The table has a name, REPRESENTATIVE, and the six vertical columns each have names (case insensitive in Oracle) and each column stores a particular kind of data[2]. These names are carefully selected to describe the table or column, just as you carefully selected meaningful and descriptive names for variables, constants, user-defined functions, procedures, and objects such as command buttons in the

[1] I would very much like to thank Professor Rogler for allowing me to provide this reference material to students completing CS 83 R.

[2] Databases can have relationships between data, but a **knowledgebase** in an expert system or chatbot more intimately links data. In philosophy, epistemology is "the study or theory of the origin, nature, methods, and limits of knowledge." **Cognitive science** combines the fields of computer science, philosophy, psychology, and neurology to study how knowledge is stored and accessed in biological brains, artificial neural networks, and conventional (Turing) computers.

computer languages you've studied. These columns store the properties or attributes of one of the entities.

The raw data in this table is presented in five horizontal rows, a row for each representative. Already you can see that the database is more than the values in those rows, for the table and column names are part of the database. Also part of the database are the datatype for each column—four columns of text with columns two and three limited to 20 characters and column four with two characters, a column of dates, and another column of text. While the first column appears here as integers, actually they're being stored digit by digit as a string. The values for the REPRESENTATIVE_ID are not just any digits. They are unique and lie in the range of (say) 01 to 99. Their uniqueness and their range are enforced during data entry by **data validation rules.** These rules are also stored as part of the database.

I'll next review simple ideas on how data is stored in a file (even a text file)—ideas that you've perhaps used in some programming course.

Sometimes files of data include special characters called **delimiters** that separate the various fields and various records. Delimiters are often commas, semicolons, tabs, or quotation marks between fields and a carriage return or combination carriage return/line feed (ASCII characters 13 and 10) between records.

Comma-delimited files or files with other delimiters can be **exported** from a DBMS (a Database Management System program such as Oracle, SQL Server, Access, or MySQL) as a text file. **Text files** are files of ASCII characters (or other collating sequences such as Unicode) that can be viewed, edited, and saved from an ordinary text editor such as Notepad. But note that other data exists in a database other than this data output to a text file—data such as table names and column names, datatypes designating how data will be stored in a column, data validation rules—so information can be lost in an export.

For some databases, the delimiters allow the fields to have different lengths in different records. Such a database can have **variable-length records**. In fact, a field can be empty. For example, if the field delimiter is a <tab>, then <tab><tab> indicates that no information is stored for the field.

Some files of data have a fixed structure. If the corresponding fields of various records have the same lengths, then that table has **fixed-length records**. In such a database, no delimiters between fields or records are necessary. If the size of the fields are fixed and known, then the computer can calculate an **offset** (perhaps measured in bytes) from the first record to reach any other record or field.

To search sequentially through a variable-record-length database can be time consuming because of the many tests for delimiters necessary to determine which field of which record is being accessed. But most of this run-time burden can be eliminated by **indexing** the list of data so the testing for the delimiters and determining the addresses of the various fields and records is done once and the results stored in a table. Thereafter, the index is accessed to determine the address of each field (or at least the address of each record). The index table can also pre-record any arrangement (ascending or descending alphabetical or numerical value) of any field so searching and sorting records based on that field is much faster.

Another characteristic of databases is how the data is stored. One way is to store every letter, decimal digit, punctuation mark, or other character as an ASCII character or some other collating sequence in one byte or two bytes[3] each. This approach requires that conversions take place between the ASCII characters for numbers and the binary forms used internally in the computer for arithmetic.

[3] One 8-bit byte can store the decimal numbers 0 to 255, or a total of 256 different numbers, each which can correspond to a different character as in the ASCII-8 character set or EBCDIC set or other collating sequence with

Even text need not be stored as one byte per character, for it may have been **compressed** by one of many schemes. One compression scheme is to assign a number in the range 0 to 65,535 to the 65,536 most popular words. If the word *hippopotamus* were assigned the number 4096, that integer could be stored in the two bytes with bits 00010000 00000000 rather than 12 separate bytes for *hippopotamus.*

Another way to store numbers is the binary form native for that data, where an integer in the range –32768 to 32767 is stored in its two-byte (16-bit) binary form, a floating point number such as -0.12345×10^4 is stored in its binary form with parts of the storage used for the sign (-), mantissa (0.12345), and exponent (4).

Of course, information includes much more than the text and numbers we read from the printed page. Faxes, sounds, still and animated graphical images, and digital video each have different formats. The binary data of a movie in digital format can be a field. With a binary field as large as four gigabytes, one movie can be one field in an Oracle database. More commonly, such huge fields are stored outside the database under the control of the operating system, but with pathnames (drive, folders, and filenames) to these external files stored inside the database.

Access to the data can be sequential or random. **Sequential access** means that the database file is accessed record by record beginning with the first. **Random access** means that the records can be accessed in any order or it can be accessed sequentially also since the order 1, 2, 3, … is a possible order of access.

In the case of a file being sequentially accessed with some 3rd Generational Language, you read each record by reading a series of variables (the fields that comprised the record). Fields are separated by a comma (or another delimiter) and records are separated by a <Carriage Return> (or another end-of-line delimiter). Any strings begin and end with a double-quote mark ("). Conversions between the native data types as stored in memory and the ASCII data in the file were automatic.

In the case of a random access file, you write and read records in random order in the native binary format of the data (without conversions necessary between the native format and what was written in the disk). After a record was read, you sort out the fields by using the RecordName.FieldName reference.

With many computer languages, you can retrieve data from relational databases such as Oracle.

Oracle, IBM DB2, Microsoft SQL Server, and the popular PC databases such as dBase, FoxPro, Paradox, and Access are **relational database management systems (RDBMS)**. The precise definition of a RDBMS is based on relational mathematics, and we will study its rules later. For now, a RDBMS can link (join) two or more tables so data can be extracted from one or more tables. Each table is composed of columns and rows. A row is a record, and a record has one or more fields. For now, a relation is a special table, one that has a unique name in the database, each column has a unique name in the table it is part of, and each column has a datatype. There are other rules we'll see later.

A relational database can link (**join**) two or more tables if the tables are properly designed. What links two tables is that a column (or maybe more than one column) in a table references or links to a column (or more than one column) in a table. For example on the next page are four tables modified from Capron, 6th edition, with lists of the orders, customers, inventory, and sales representatives.

---

256 characters. Two bytes or 16 bits can store the decimal numbers 0 to 65,545 or 65,546 different numbers, each which can represent a different character. One form of Unicode uses two bytes to store each character.

The first two tables (INVOICE and CUSTOMER) are linked by the CUSTOMER_NUMBER field, the first and third tables (INVOICE and INVENTORY) are linked through the ITEM_NUMBER field, and the 2nd and 4th tables (CUSTOMER and REPRESENTATIVE) are linked through the REPRESENTATIVE_ID field.

The four tables are not separate files but are integrated into one file perhaps along with other pieces such as instructions to generate reports, queries, forms, macros, etc. The data and instructions are manipulated with the database management system (DBMS).

INVOICE table

INVOICE_ NUMBER	CUSTOMER_ NUMBER	INVOICE_ DATE	ITEM_ NUMBER	QUANTITY
01	20	12-MAY-99	70	11
02	30	28-FEB-99	60	15
03	30	13-SEP-00	20	14
04	20	10-JUL-01	10	10
05	60	31-AUG-01	60	20
Primary key	Foreign key, NOT NULL		Foreign key	
Varchar2(2)	Varchar2(2)	Date	Varchar2(2)	Number(2)

CUSTOMER table

CUSTOMER_ NUMBER	CUSTOMER_NAME	CITY	REPRESENTATIVE_ID
10	Ballard Computer	Seattle	55
20	Computer City	Miami	33
30	Under_Score, Inc.	Atlanta	22
40	Varner User System	Naperville	44
50	100% Jargon	Spokane	55
60	Computing Solutions	Tucson	11
Primary key	NOT NULL		Foreign key
Varchar2(2)	Varchar2(20)	Varchar2(20)	Varchar2(2)

INVENTORY table

ITEM_NUMBER	DESCRIPTION	QUANTITY_ON_HAND
10	Hand Scanner	191
20	Modem	453
30	Hard Drive	294
40	Printer pack	676
50	CD-ROM drive	817
60	3 1/2" disk holder	982
70	Sound card	0
80	Mouse	296
90	Rogler's DSL	152
Primary key		NOT NULL
Varchar2(2)	Varchar2(20)	Number(3)

REPRESENTATIVE table

REPRESENTATIVE_ID	LAST_NAME	FIRST_NAME	REGION	HIRE_DATE	PHONE
11	Rogler	Harold	SW	05-JAN-99	(310) 456-7890
22	Higgins	Heather	SE	16-DEC-91	(404) 524-8472
33	Sullivan	Pat	NE	21-FEB-88	(305) 734-2987
44	Speed	Kristen	MW	14-JUN-90	(708) 823-8222
55	Sigafoos	Alex	NW	05-MAR-01	(310) 123-7890
Primary key	NOT NULL		CHECK		
Varchar2(2)	Varchar2(20)	Varchar2(20)	Char(2)	Date	Varchar2(14)

Structured Query Language, SQL

SQL, pronounced S.Q.L. or see' quel, is a language to access (retrieve) information in relational databases. SQL is also used to create and alter the structure of a table, and to insert, update, or delete data, and many other things. SQL is non-procedural: it contains no instructions on how to access and select the data. How to access and select the data is handled internally by SQL and is hidden from the user.

An SQL query for the INVENTORY table above is

```
SELECT DESCRIPTION, QUANTITY_ON_HAND
  FROM INVENTORY
  WHERE ITEM_NUMBER = '10';
```

SQL is a **standard language** that exists as an American National Standards Institute (ANSI) standard and as an International Standards Organization (ISO) standard. Oracle and other developers of relational DBMS add enhancements to the standard SQL, and Oracle's version is called **SQL*PLUS**. In CS60, SQL is introduced in Chapter 07. With Microsoft Access, you can query a database either with its Query By Example graphical interface or by using SQL.

Oracle's PL/SQL language

PL/SQL includes the selection structure in programming (e.g., IF-THEN-ELSE statement), repetition structure (loops), variables, and datatypes for records and arrays. User-defined functions, procedures, triggers (programs that run automatically when a database is changed), and other features are possible with PL/SQL. Some SQL statements can be embedded (included) in PL/SQL.

An example of **a selection statement** from PL/SQL is

A *condition that is **True** or **False***

```
IF quantity_ordered <= 10 THEN
   price := 5.0;
ELSE
   price := 4.0;
END IF;
```

An example of **a loop** from PL/SQL is

```
FOR i IN 1..100 LOOP
   sum := sum + i;
END LOOP;
```

The details of these statements are unimportant now. In CS60, PL/SQL is introduced in Chapter 09. Not all relational databases have such selection and repetition statements to manipulate the data.

Visual Basic, C, COBOL, Java, and other languages also can be used to access data in databases.

The parts of a database management system

A **database management system** (DBMS) is a program that allows you to

- set up and revise the structure of the database (including the number of tables and their names, number of columns, datatypes of the columns, names of the columns, constraints such as keys)
- add records to a table and delete records from a table. One way to do these is with a **form**, a display on a screen of one or more records or parts of those records. The data entry operator can edit the displayed values, create new rows, and enter data into them. Another way is to use SQL or PL/SQL.
- update (change) the values in the records.
- access existing data to respond to queries. A **query** is a question or instruction to retrieve data from tables. A query can be posed in SQL.
- access existing data to generate **reports**, output intended to be printed.
- manipulate the data to produce new data.
- provide masks to help the user enter phone numbers in the proper format.
- validate—test the data for its accuracy (does the data lie in a certain range, is the data an integer or some other data type),
- control whether an operator must supply entries for certain fields or whether a field can be left empty (does the data allow nulls?)
- trigger a change to one table as a result of a change to another (e.g., decrease the quantity on hand as a result of a sale). Programs called triggers are described in Chapter 10.

A diagram for a generic database management system (DBMS) is shown below.

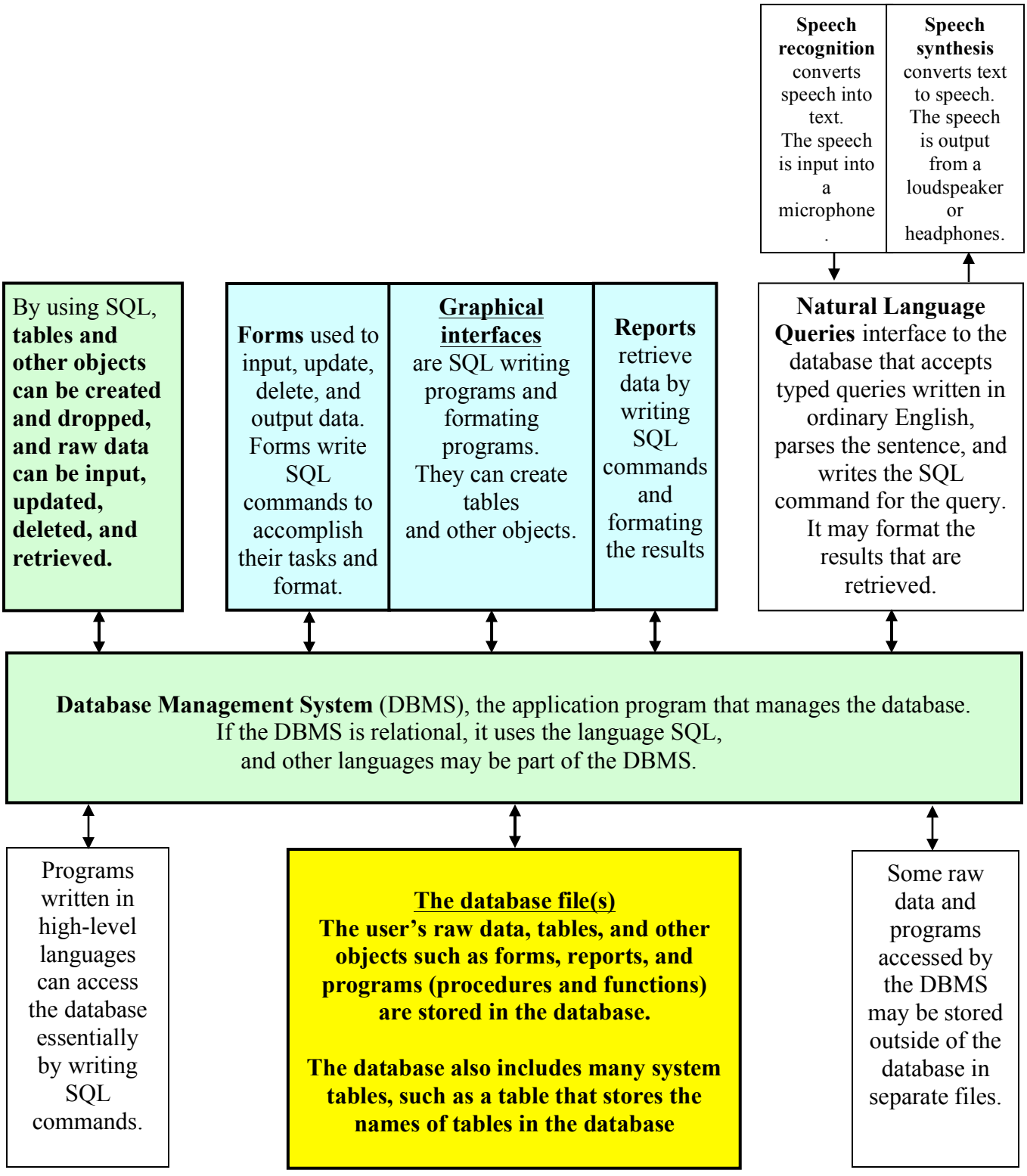

Figure 1.1. A generic database management system with its data and metadata

The data files created with many computer languages are composed of the raw data, and separate programs are used to store, modify, and view the data. However, a RDBMS integrates all raw data, table structures, index tables, data validation instructions, and instructions for forms, reports, and queries into one file or a few files.

Steps to design a relational database

Many problems with databases are a result of their design—enough problems to warrant separate books on designing relational databases. In brief, the steps to design an RDBMS are:

1. Identify the information the user will get from the database. This identifies the fields of the table(s).

2. Organize the fields into one or more tables. Group the fields that are logically related into the same table. Eliminate or reduce redundancies by satisfying normalization rules that we will study in Chapter 04. Specify a datatype for each column (we will study data types in Chapter 06). Specify a primary key for each table.

3. If the database contains more than one table, establish relationships between the tables by specifying a foreign key that links to a primary key. This means to relate the tables or link the tables through common fields.

4. Create any necessary **indexes** for the table. An index allows the user to quickly search through the field that is indexed.

   An index has two parts:

   (1) the **index field** which has the values of the field or fields being indexed. An example of a useful index field on a big table of names is the Last name, First Name, Middle name.
   (2) a **pointer field** with the record number or address for each record.

   A table can have no fields indexed, one indexed, or more than one (if the table has more than one field). All fields could be indexed. In Oracle, an index is an object.

5. Define any data validation rules to ensure, as best as possible, that the data entered is accurate. Perhaps the data entered must be numeric in a certain range such as greater than 5 but less than 100, or with a certain format as for a phone number.

Tools to help design a relational database include:

(1) Paper and pencil, either freehand or using plastic templates, a ruler, and eraser.
(2) The drawing tools in Word, Powerpoint, and Visio Studio.
(3) Software programs that help design relational databases[4]

[4] Examples for Oracle are (1) Oracle's Designer by Oracle, which is the central topic in the course CS66, Advanced Oracle Programming at Santa Monica College, (2) ERwin by Logic Works, and (3) S-Designor by Sybase.

Key Terms

column
data compression
data item
data validation rules
database
database administrator (DBA)
database management system (DBMS)
database tables
datatype
delimiters between fields and records
designing a relational database
detailed and summary reports
field
fixed-length and variable-length records
forms
index field
pointer field
primary index
queries
random access and sequential access
record
relational database
relational DBMS (RDBMS)
relational operators
report
row
secondary index
table

