Relational Database Schema

Relational database schema

- Can also be referred to as a “data dictionary”
- Defines the data types of each field of each table
- Can also include format information e.g., a date can have a format like DD-MM-YYYY

Data types

TABLE 7.4

SOME COMMON SQL DATA TYPES

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D) or NUMERIC(L,D)	The declaration NUMBER(7,2) or NUMERIC(7,2) indicates that numbers will be stored with two decimal places and may be up to seven digits long, including the sign and the decimal place (for example, 12.32 or –134.99).
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a *minimum* specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as *Smith* and *Katzenjammer* are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) or VARCHAR(25) will let you store characters up to 25 characters long. However, unlike CHAR, VARCHAR will not leave unused spaces. Oracle automatically converts VARCHAR to VARCHAR2.
Date	DATE	Stores dates in the Julian date format.

INTEGER types:

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

STRING types:

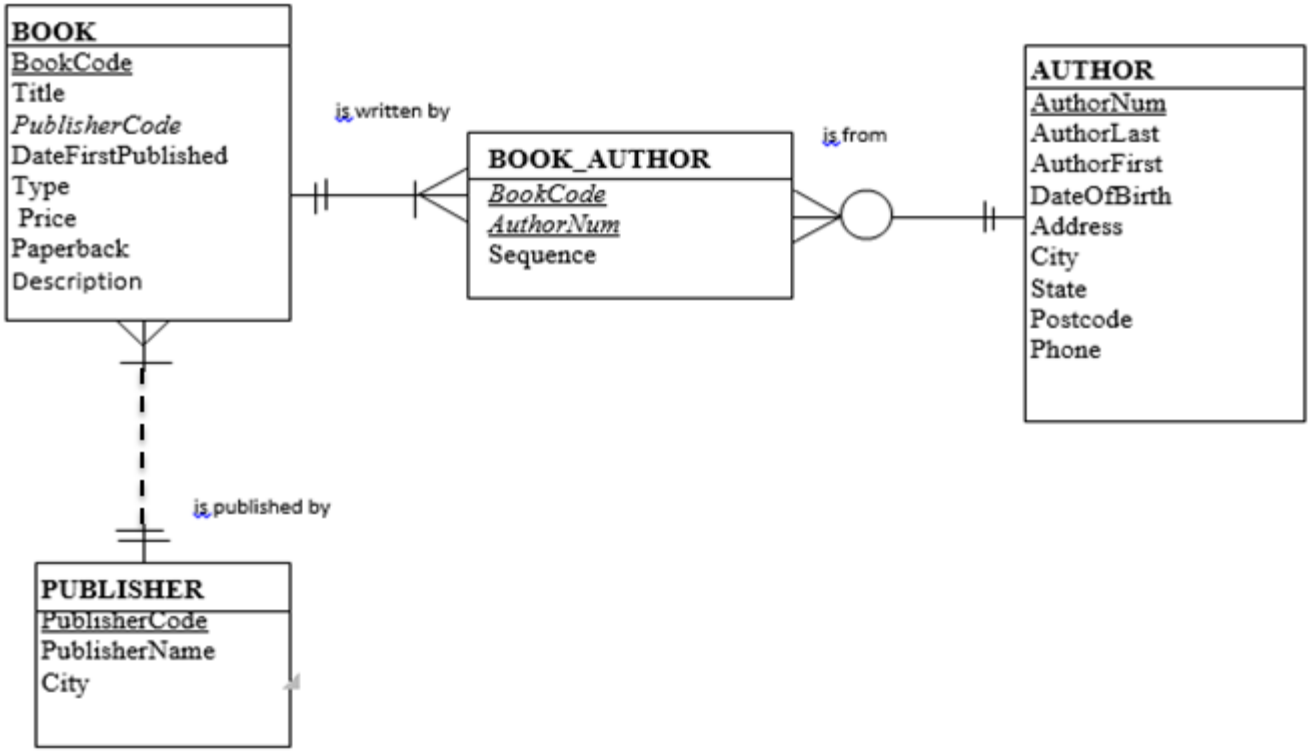
Value	CHAR(4)	Storage Required	VARCHAR(4)	Storage Required
''	' '	4 bytes	''	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

Learn more from (must do):

- https://dev.mysql.com/doc/refman/5.7/en/data-types.html
- https://www.tutorialspoint.com/mysql/mysql-data-types.htm

Relational database schema

- Let's we are given an ERD and it's relation schema
- Can you check if all tables are in 3NF?



Relational database schema

Table Name	Field	Type	Description
BOOK	BookCode	CHAR(6)	PRIMARY KEY
	Title	VARCHAR(40)	
	PublisherCode	CHAR(3)	FOREIGN KEY REFERENCES PUBLISHER(PublisherCode)
	DateFirstPublished	DATE	**Format:** DD-MM-YYYY
	Type	CHAR(3)	
	Price	DOUBLE	
	Paperback	CHAR(1)	
	Description	VARCHAR(30)	
AUTHOR	AuthorNum	INT (11)	PRIMARY KEY NOT NULL AUTO_INCREMENT
	AuthorLast	VARCHAR(12)	
	AuthorFirst	VARCHAR(10)	
	DateOfBirth	DATE	
	Address	VARCHAR(30)	
	City	VARCHAR(30)	
	State	CHAR(3)	
	Postcode	CHAR(4)	
	Phone	VARCHAR(15)	
BOOK-AUTHOR	BookCode	CHAR(6)	PRIMARY KEY FOREIGN KEY REFERENCES BOOK(BookCode)
	AuthorNum	INT(11)	PRIMARY KEY FOREIGN KEY REFERENCES AUTHOR(AuthorNum)
	Sequence	INT	
PUBLISHER	PublisherCode	CHAR(3)	PRIMARY KEY
	PublisherName	VARCHAR(25)	
	City	VARCHAR(30)	

- For columns with fixed length values (e.g., Postcode, BookCode) use CHAR(…) instead of VARCHAR(…).
- CHAR saves memory for fixed length values (e.g., s-number), VARCHAR saves memory of variable length values (e.g., name).

Thank you