



# Default Values

```
CREATE TABLE SAILOR (
  Sid [sidtype] [sidsign] [sidnull],
  Sname [sname] [sname] [sname],
  Rating [ratingtype] [ratingsign] [ratingnull],
  Age [age] [age] [age],
  [pk])
```

SAILOR(Sid, Sname, Rating, Age)

Sid	Sname	Rating	Age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Sid	NOT NULL	5 respondents	83 %		✓
	DEFAULT NULL		0 %		
Sname	NOT NULL	4 respondents	67 %		
	DEFAULT NULL	2 respondents	33 %		✓
Rating	NOT NULL	4 respondents	67 %		
	DEFAULT NULL	2 respondents	33 %		✓
Age	NOT NULL	3 respondents	50 %		
	DEFAULT NULL	3 respondents	50 %		✓

Other points –

- Sid is NOT NULL because it is the primary key
- Sname, Rating, Age are not required, so DEFAULT NULL is appropriate
- use NOT NULL for required attributes

# Primary Key

```
CREATE TABLE SAILOR (
  Sid [sidtype] [sidsign] [sidnull],
  Sname [sname] [sname] [sname],
  Rating [ratingtype] [ratingsign] [ratingnull],
  Age [age] [age] [age],
  [pk])
```

SAILOR(Sid, Sname, Rating, Age)

Sid	Sname	Rating	Age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

[nothing]	1 respondent	17 %		
PRIMARY KEY (Sid)	5 respondents	83 %		✓

# MySQL Data Types

RESERVATION(Sid, Bid, Day)

Sid	Bid	Day
22	101	1998-10-10
22	102	1998-10-10
22	103	1998-10-08
22	104	1998-10-07
31	102	1998-11-10
31	103	1998-11-06
31	104	1998-11-12
64	101	1998-09-05
64	102	1998-09-09
74	103	1998-09-08

Day

CHAR(10)	1 respondent	17 %		
VARCHAR(10)	1 respondent	17 %		
BINARY(10)		0 %		
VARBINARY(10)		0 %		
TINYBLOB		0 %		
TINYTEXT		0 %		
DATE	4 respondents	67 %		✓
NOT NULL	5 respondents	83 %		
DEFAULT NULL	1 respondent	17 %		✓
[nothing]		0 %		

- DATE is most appropriate
- VARCHAR uses more space than CHAR when fields are all the same length (or close to the same length)
- TEXT is intended for larger quantities of text

Other points –

- Day is not required, so DEFAULT NULL is appropriate
- use NOT NULL for required attributes

# Primary and Foreign Keys

RESERVATION(Sid, Bid, Day)

- RESERVATION.Sid -> SAILOR.Sid
- RESERVATION.Bid -> BOAT.Bid

Sid	Bid	Day
22	101	1998-10-10
22	102	1998-10-10
22	103	1998-10-08
22	104	1998-10-07
31	102	1998-11-10
31	103	1998-11-06
31	104	1998-11-12
64	101	1998-09-05
64	102	1998-09-09
74	103	1998-09-08

```
CREATE TABLE RESERVATION (
  Sid smallint(5) unsigned NOT NULL,
  Bid smallint(5) unsigned NOT NULL,
  Day [daytype] [daynull],
  [pk],
  [fk],
  [fk])
```

[nothing]		0 %		
PRIMARY KEY(Sid, Bid)	2 respondents	33 %		
PRIMARY KEY(Sid, Bid)	4 respondents	67 %		✓

• Bid, Sid together are the primary key

[nothing]	3 respondents	50 %		
CONSTRAINT fk1 FOREIGN KEY (Bid) REFERENCES BOAT (Bid) ON UPDATE CASCADE, CONSTRAINT fk2 FOREIGN KEY (Sid) REFERENCES SAILOR (Sid) ON UPDATE CASCADE	3 respondents	50 %		✓

• Bid and Sid are separate FKs (they refer to different tables)

[nothing]		0 %		
KEY fk (Bid, Sid)		0 %		
KEY fk1 (Bid), KEY fk2 (Sid)	6 respondents	100 %		✓

• KEY defines indexes – needed for foreign keys

- Bid and Sid are separate FKs

CPSC 343: Database Theory and Practice • Fall 2024 9

## Constraints

```

BOOK (Book_id, Title, Publisher_name)
BOOK_AUTHORS (Book_id, Author_name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_id, Branch_id, No_of_copies)
BOOK_LOANS (Book_id, Branch_id, Card_no, Date_out, Due_date)
LIBRARY_BRANCH (Branch_id, Branch_name, Address)
BORROWER (Card_no, Name, Address, Phone)
  
```

The name uniquely identifies publishers.	primary key constraint
By default, the number of copies of a book at a branch is 1.	DEFAULT
The number of copies can't be negative.	unsigned
The due date cannot be before the date the book was checked out.	CHECK constraint
The combination of name and phone number also uniquely identify particular borrowers.	UNIQUE
Phone numbers must be of the form XXX-XXX-XXXX.	CHECK constraint
Only borrowers that are in the database can check out books.	foreign key constraint
By default, books are due two weeks after they are checked out.	trigger- DEFAULT

UNIQUE also specifies no duplicates, but PK is the best match for *uniquely identifies*

not taking space for unnecessary sign (unsigned) is better than adding another check to perform (CHECK constraint)

not FK – borrower Name and Phone are only attributes of BORROWER – no reference to another table  
not “can't be enforced” – ER diagrams, relational schemas don't have a notation for additional keys but this info can be stated separately

not column data type – can't be more specific than CHAR(10) or CHAR(12)

prefer FK to triggers when possible (borrowers are in BORROWERS and are referenced in BOOK\_LOANS)

CHECK constraints can only be simple conditions – no subqueries

expressions are allowed for DEFAULT in MySQL 8+ prior to MySQL 8+ allowed only constant literals and CURRENT\_TIMESTAMP

CPSC 343: Database Theory and Practice • Fall 2024 10

## Constraints

```

BOOK (Book_id, Title, Publisher_name)
BOOK_AUTHORS (Book_id, Author_name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_id, Branch_id, No_of_copies)
BOOK_LOANS (Book_id, Branch_id, Card_no, Date_out, Due_date)
LIBRARY_BRANCH (Branch_id, Branch_name, Address)
BORROWER (Card_no, Name, Address, Phone)
  
```

A book's title is required.	NOT NULL
By default, the check out date for a book is the date on which the BOOK_LOANS entry is recorded.	DEFAULT
The card number is an alphanumeric string (can contain letters and digits).	column data type
A particular borrower can't have more than 5 books checked out at a time.	trigger
A borrower can't check out more than two books on the same day.	trigger
A book must have at least one author recorded.	trigger can't be enforced

not PK – PK does imply NOT NULL, but it is primarily about uniquely identifying rows

DEFAULT allows functions like CURDATE() for MySQL 8+  
not FK – there's not another table with valid dates for Date\_out; “default value” is about what happens when an INSERT doesn't include Date\_out

CHAR/VARCHAR is the best (but not exact) match for the column data type  
CHECK constraint is also valid (can specify regex)

not CHECK constraint – CHECK constraints can only involve a single row (and no subqueries)

not NOT NULL – the constraint says that an entry in BOOK requires an entry in BOOK\_AUTHOR; the FK BOOK\_AUTHORS.Book\_id → BOOK.Book\_id goes in the wrong direction

can't be enforced – the constraint itself creates a circular dependency with the FK in BOOK\_AUTHOR; while a trigger can insert an author when a book is inserted, what name should be used?

CPSC 343: Database Theory and Practice • Fall 2024

## Questions

What is the difference between “can't be enforced” and a CHECK constraint?

- “can't be enforced” means there isn't a way to enforce the requirement
  - e.g. a book must have at least one author
    - BOOK doesn't have author info, so can't use NOT NULL – what column would it apply to?
    - must insert book before any authors (due to FK BOOK\_AUTHORS.Book\_id → BOOK.Book\_id) so can't deny BOOK insertion, and no way to require a subsequent insertion of an author
- a CHECK constraint allows constraints on column values
  - e.g. Due\_date >= Date\_out

```

BOOK (Book_id, Title, Publisher_name)
BOOK_AUTHORS (Book_id, Author_name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_id, Branch_id, No_of_copies)
BOOK_LOANS (Book_id, Branch_id, Card_no, Date_out, Due_date)
LIBRARY_BRANCH (Branch_id, Branch_name, Address)
BORROWER (Card_no, Name, Address, Phone)
  
```

CPSC 343: Database Theory and Practice • Fall 2024 12

## Questions

What's the difference between a trigger and just using INSERT, DELETE, UPDATE?

- the trigger performs an operation in addition to the insert/delete/update operation
- can be used to check/enforce complex constraints not possible by other means
- can be used to perform other bookkeeping
  - not limited to checking constraints

## Questions

When should you use a trigger instead of a CHECK (UNIQUE, NOT NULL, or other) constraint?

- use a trigger when other mechanisms are insufficient
  - e.g. a CHECK constraint cannot involve a subquery

```
Students are not allowed to enroll in more than four courses:
CREATE TRIGGER maxenroll
BEFORE INSERT
ON EnrolledIn FOR EACH ROW
BEGIN
  IF ( SELECT COUNT(*)
      FROM EnrolledIn
      WHERE studentid = NEW.studentid ) >= 4
  THEN SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Cannot enroll in more than four courses'
  END IF
END
```

The error code '45000' is a generic value meaning "unhandled user-defined exception".