

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 – foreign key references a specific row in another table

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

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?

CHECK Constraints

```
ALTER TABLE table
ADD CONSTRAINT name CHECK (condition);
```

```
ALTER TABLE table
DROP CONSTRAINT name;
```

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
 - prefer other mechanisms because a trigger is only run on an INSERT, UPDATE, DELETE operation
- a trigger can also be used to perform other bookkeeping
 - not limited to checking constraints

Triggers

```
CREATE TRIGGER name  
when event  
ON A FOR EACH ROW  
action
```

when is BEFORE or AFTER
event is INSERT, DELETE, UPDATE

```
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".

OLD refers to the existing row
NEW refers to the row after the operation

need BEGIN...END if there are multiple statements in the trigger body