

Database Implementation

Physical Database Design

Covers –

- implementing the schema
 - translating relations into CREATE TABLE statements
 - choosing concrete SQL data types for each attribute
 - declaring constraints
 - choosing surrogate keys vs natural keys
- storage and access structures
 - choosing a storage engine
 - selecting indexes
 - considering how PK choice affects physical organization
- performance-driven decisions
 - denormalization as a deliberate tradeoff
 - partitioning or splitting large tables
 - estimating data volume and query workload

Physical Database Design

Covers – (continued)

- security and administration
 - defining users, roles, and privileges
 - views for restricting or simplifying access
 - character sets
 - backup and recovery considerations
- other implementation concerns
 - database objects (views, triggers, stored routines)
 - enforcing constraints the logical model couldn't express

Key Points – Data Definition

- GUI tools like MySQL Workbench may provide interfaces for creating tables but having knowledge of the SQL is important
 - GUI may not support everything you need

CREATE TABLE

- CREATE TABLE maps relations into tables
 - for each column – name, data type, (optional) column constraints
 - foreign keys can specify what happens on delete/update
- the order of table creation matters
 - tables referenced by foreign keys must be created first

```
CREATE TABLE LOAN (
  loan_id      INT AUTO_INCREMENT,
  borrower_id  INT NOT NULL,
  copy_id      INT NOT NULL,
  checkout_date DATE NOT NULL DEFAULT (CURRENT_DATE),
  due_date     DATE NOT NULL,
  PRIMARY KEY (loan_id),
  CONSTRAINT fk_loan_borrower
    FOREIGN KEY (borrower_id) REFERENCES BORROWER (borrower_id)
    ON DELETE RESTRICT,
  CONSTRAINT fk_loan_copy
    FOREIGN KEY (copy_id) REFERENCES BOOKCOPY (copy_id)
    ON DELETE RESTRICT,
  CONSTRAINT chk_loan_dates CHECK (due_date > checkout_date)
);
```

5

CREATE TABLE

```
CREATE TABLE tbl_name (
  col_name data_type [NOT NULL | NULL]
    [DEFAULT {literal | (expr)}]
    [AUTO_INCREMENT],
  ...

  PRIMARY KEY (col_name [, col_name] ...),

  [CONSTRAINT symbol] UNIQUE (col_name [, col_name] ...),

  [CONSTRAINT symbol] FOREIGN KEY (col_name [, col_name] ...)
    REFERENCES parent_tbl (col_name [, col_name] ...)
    [ON DELETE reference_option]
    [ON UPDATE reference_option],

  [CONSTRAINT symbol] CHECK (expr)
);

reference_option:
  RESTRICT | CASCADE | SET NULL | NO ACTION
```

CPS 343: Database Theory and Practice • Fall 2026

6

MySQL Data Types

Type	Storage (Bytes)	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	127	255
SMALLINT	2	32767	65535
MEDIUMINT	3	8388607	16777215
INT	4	2147483647	4294967295
BIGINT	8	2 ⁶³ -1	2 ⁶⁴ -1

Key points –

- choose the smallest-size data type that is sufficient for the data requirements
 - INT is the typical choice for keys and counts
 - use TINYINT/SMALLINT for small bounded values (ratings, quantities)
 - use TINYINT for small flags / enumerated values – BOOLEAN for true/false
- add UNSIGNED to disallow negatives and double the positive range
- AUTO_INCREMENT applies to integer columns and is the usual way to generate surrogate keys
- INT(w) – w is the display width, not a range of legal values
- DECIMAL(p,s) – p (precision) is the total number of digits, s (scale) is the number of digits after the decimal point
- DECIMAL allows exact representation of s decimal points; FLOAT/DOUBLE do not
 - use DECIMAL for prices, currency
 - use FLOAT/DOUBLE for measurements

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

TINYINT is sufficient for this data set, but the limit of 255 sailors might not be enough in general

TINYINT(3)
SMALLINT(3)
MEDIUMINT(3)
INT(3)
BIGINT(3)
DECIMAL(5,0)
DECIMAL(5,0)
DOUBLE(5,0)

Sid

TINYINT(3)
SMALLINT(3)
MEDIUMINT(3)
INT(3)
BIGINT(3)
DECIMAL(2,0)
DECIMAL(2,0)
DOUBLE(2,0)

Rating

TINYINT(4)
SMALLINT(4)
MEDIUMINT(4)
INT(4)
BIGINT(4)
DECIMAL(3,1)
DECIMAL(3,1)
DOUBLE(3,1)

Age

Other points –

- Sid, Rating should be UNSIGNED

MySQL Data Types

Key points –

- CHAR/VARCHAR are for text
 - CHAR is fixed length – CHAR(L) takes up L bytes (max 255 chars)
 - VARCHAR is variable length – VARCHAR(L) takes up data+1 or data+2 bytes (must also store the actual length) (max 65,535 bytes per row)
 - use VARCHAR for names, emails, titles, usernames
 - use CHAR for fixed-format codes e.g. state, ISBN-13, currency code
 - use CHAR/VARCHAR for password hashes if stored as encoded text
- BINARY/VARBINARY are for binary data
 - use for small raw values e.g. password hashes (if stored as raw bytes), UUIDs, encrypted data, checksums
- TEXT/BLOB are intended for larger quantities of freeform text/binary data
 - differences in how indexing is done for TEXT vs CHAR/VARCHAR
 - use TEXT for long descriptions or comments, CHAR/VARCHAR for short searchable fields
 - use BLOB for images, documents though often better to only store the path in the DB

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

Sname

CHAR(45)
VARCHAR(45)
BINARY(45)
VARBINARY(45)
TINYBLOB
TINYTEXT

TINYTEXT/BLOB – 255 bytes
TEXT/BLOB – 65,535 bytes (64KB)
MEDIUMTEXT/BLOB – 16MB
LONGTEXT/BLOB – 4GB

8

MySQL Data Types

Key points –

- DATE stores date only – 3 bytes
range 1000-01-01 to 9999-12-31
- TIME stores time of day or a duration – 3 bytes + fsp
range -838:59:59 to 838:59:59
- DATETIME stores date and time – 5 bytes + fsp
range 1000-01-01 00:00:00 to 9999-12-31 23:59:59
- TIMESTAMP stores data and time, time zone aware – 4 bytes + fsp
range 1970-01-01 00:00:01 to 2038-01-19 03:14:07 UTC
- YEAR stores a 4-digit year – 1 byte
range 1901 to 2155
- fsp = fractional seconds precision 0-3
additional bytes for extra precision e.g. DATETIME(3) for milliseconds

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

- 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

Column Constraints

- constraints that apply to individual columns of a table
 - PRIMARY KEY
 - FOREIGN KEY
 - NOT NULL
 - UNIQUE
 - DEFAULT
 - CHECK
- constraint names must be unique within the database
 - applies to FOREIGN KEY, CHECK

PRIMARY KEY

- the primary key uniquely identifies each row in the table
- implies NOT NULL and UNIQUE
- only one primary key is allowed per table
 - can be composite (span several columns)
- often combined with AUTO_INCREMENT to generate surrogate IDs

```
CREATE TABLE BORROWER (
    borrower_id INT AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    PRIMARY KEY (borrower_id)
);

CREATE TABLE BOOK_AUTHOR (
    isbn CHAR(13),
    author_id INT,
    PRIMARY KEY (isbn, author_id)
);
```

FOREIGN KEY

- links a column in a child table to a key in a parent table, enforcing referential integrity
 - can be composite
- the referenced column must be a PK or otherwise indexed (usually UNIQUE)
- column types must match
- naming the constraint with CONSTRAINT fk_name makes error messages readable and lets you drop it later
- ON DELETE / ON UPDATE actions specify what happens if a parent table row is deleted or updated
 - RESTRICT / NO ACTION – block the change (default)
 - CASCADE – delete or update the child rows too
 - SET NULL – set the child column to NULL (column must allow NULLs)

```
CONSTRAINT fk_loan_borrower
FOREIGN KEY (borrower_id) REFERENCES BORROWER (borrower_id)
ON DELETE RESTRICT
ON UPDATE CASCADE
```

(a clause of the CREATE TABLE statement)

NOT NULL

- requires the column to always have a value
 - inserting a row without a value (and no default) causes an error
- declared on the column itself, not as a separate table constraint
- PK columns are automatically NOT NULL

```
name      VARCHAR(100) NOT NULL,  
email     VARCHAR(255) NOT NULL,  
phone     VARCHAR(20)      -- nullable
```

(clauses of the CREATE TABLE statement)

UNIQUE

- no two rows may share the same value in the column(s)
 - can be composite
- a table can have many unique constraints
- NULLs are allowed and multiple NULLs may occur
 - NULL is not equal to NULL
- automatically creates an index

```
-- isbn and copy_number are each globally unique  
CONSTRAINT uq_isbn UNIQUE (isbn),  
CONSTRAINT uq_copy_number UNIQUE (copy_number)  
  
-- Each copy number is unique within a book  
CONSTRAINT uq_copy UNIQUE (isbn, copy_number)
```

(clauses of the CREATE TABLE statement)

DEFAULT

- supplies a value when an INSERT doesn't specify one
- declared on the column itself, not as a separate table constraint
- can be a literal or an expression in parentheses (MySQL 8.0.13+)
- an explicit NULL in an insert overrides the default

```
status      VARCHAR(10) NOT NULL DEFAULT 'active',  
max_loans   INT          NOT NULL DEFAULT 5,  
checkout_date DATE       NOT NULL DEFAULT (CURRENT_DATE),  
created_at  DATETIME     DEFAULT (CURRENT_TIMESTAMP)
```

(clauses of the CREATE TABLE statement)

CHECK

- rejects rows where the condition evaluates to false
 - disallows INSERT or UPDATE that would make that happen
 - allows rows where the condition involves NULL (evaluates to unknown)
- can compare a column to a constant or to other columns in the same row
 - within-row conditions only – cannot involve other rows in the same table or in different tables
- enforced only in MySQL 8.0.16+ – older versions parse and ignore

```
CONSTRAINT chk_loan_dates CHECK (due_date > checkout_date),  
CONSTRAINT chk_max_loans CHECK (max_loans >= 0 AND max_loans <= 10),  
CONSTRAINT chk_status CHECK (status IN ('active', 'suspended'))
```

(clauses of the CREATE TABLE statement)

Modifying CHECK Constraints

- can add/remove CHECK constraints after the fact
 - MySQL Workbench doesn't support specifying CHECK constraints – must be done manually

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

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

- existing data is checked when a constraint is added – ALTER will fail if current rows violate the constraint
- it is also possible to add/remove other constraints and to modify columns after the fact

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

Implementing 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.	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

Implementing 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.	
By default, the check out date for a book is the date on which the BOOK_LOANS entry is recorded.	
The card number is an alphanumeric string (can contain letters and digits).	
A particular borrower can't have more than 5 books checked out at a time.	
A borrower can't check out more than two books on the same day.	
A book must have at least one author recorded.	

Implementing 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.	can't be enforced

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

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_AUTHORS; the FK BOOK_AUTHORS.Book_id → BOOK.Book_id goes in the wrong direction