DBMS II - Schema Design

① Data types

- String ✓
- Numeric

② Schema design

- What data goes in which table?
- Cardinality
- Case Study

Data types

```
CREATE TABLE `users` (
    name: String → varchar
);
```

1. String
2. Numeric

5. Boolean

6. Date (Temporal) — Time

5. bLOBs | CLOB

6. JSON

7. Enums

String
- Lchar
- Lstring

String

① char(x) — x - maximum

↳ fixed length string

char(4) — ④

'abca'

'a' → char(1)

a _ _ _

' a b c d é → char(4)

↳ error / strict

↳ truncate

x - 0 - 255

MySql - strict mode

- strongly type

what can we store in char?

→ Phone number codes

→ Country code – IN | UK

→ Gender – M F

→ padded

– $\langle x$

② varchar(x) →

$x \to 0$ to 65,535

$x \to$ initial allotment

→ avg. length

Varchar → run length encoding

abc – abc

– 3abc

1/2 bytes

length of the string

(3) Text

- Variable length
- lot of use cases

→ TINY TEXT 0 - 65,535 - 255 B

→ MEDIUM TEXT - 64 KB

→ Long Text - 4 GB

Text → does not support indexes

Select * from usr

String

- Char – fixed length – IN, UK
- varchar – variable length
- Text – many data

CHAR

TINY TEXT

MEDIUM TEXT

Full text indexes – TEXT

Schema design

↳ structure

Schema → blueprint of your database

→ tables
→ fields
→ FK
→ PK

User → id

- name
- phone

Design

- entities / table
- datatypes
- Visualisation
  - class diagram
  - ER diagram

Schema Design

① Requirements

② Schema design

③ Implement.

Case Study: ReScale

① Find all entities

real world

or conceptual

objects of

interest

Student

User

Batch

How can we identify entities?

- Identify all the nouns

Cardinality

A ——— B

If there is a relation b/w A & B

Cardinality? How many A:

How many B

Husband ① —— ① Wife

one to one → 1:1

How to establish Cardinality

1 : Many relation

M
1 : n
1
Student
current_batches
1
batches
M
1

M : 1

1. 1:1 one to one
2. 1: M one to many
3. M:1 many to one
4. M: N many to many

A

B

Student

Batch

- current batch
- previous batches

M : N

Cardinality

↳ where should a field/attribute be

١

١ : ٣ ←

FK on the M side

٥١

١ : ٣

M

N - 1

Student

id	batch_id
1	1
2	1

batch

id	studentId
1	1

M : N

Student

id	batch_id
1	2
1	3

Batch

id	student_id
1	1
1	2

M : N

Separate table

Mapping table

Student ...

id	student_id	batch_id
1	1	1
2	1	2

Composite — (StudentId, batchId)

PK

id } uniqueness

1

1

Relational → relations

→ FK

[1, 2, 3]

LONG TEXT BIGINT

6:14 6:20

10:50

AUTO_INCREMENT

Cardinality - Caveat | Side problems

Schema Design

11:30

RLE - 3abc

→ 1:1 → FK on any side

→ 1:M or M:1 → FK on the M side

→ M:N

1	1
1	2

duplicate

$id_1 = id_2$

duplication

Mapping table

id	sid	sid
1	1	1
2	1	2

(SID, BID)

Caveat #1

Null values

student	mentor_id
1	2

2	3
3	NULL

student_id	no of_interview_id
2	NULL
3	NULL

Sparse table

When we expect a lot of null values

it is better to use a mapping table

Join → Caching

Caveat #2 Attributes

slower

Student

id	batch_id	start_date	end_date	pause_date

- Attributes for relation

Student_id

id	name

Sb

bid	student	start d…
		end date

Data types

- String
  - → char(x) fixed (x → max length)
  - → varchar(x) – variable (→ initial allocation)

→ Text < TINYTEXT
MEDIUM → Indexed
LONG → Full text index

⇒ Case Study

→ Schema Design

→ Requirements ⇒ Entities

→ nouns → entities

→ Entities ⇒ Table

→ Identify relations

→ Cardinality

Howmany of EntityA
: How many of EntityB

→ placement of FK

1:1 → on any side

1:M → M side

M:1 ✓

M:N ✓ → mapping table

↓

Caveats

→ sparse → MT ✓

→ alot of attributes → MP ✓

on the relation

Assignment

→ Netflix

<

→ Identify entities

→ Schema design

SQL :→ CRUD

CREATE TABLE students (

PRIMARY KEY (`id`, `name`)

n of ayr.

1. MySql - DBMS
2. MySql wb - Client

1. Req
2. Ask qs on req
3. Identify entities
4. Relations

String

① char(x) → max length

- fixed length string

char(n) 0-255 B

↳ "abcd" 0-65,535

↳ "a---"

↳ "ab--

↳ "abcd" → abcd

↳ error → sql strict mode

↳ truncate → loose

(2) variable length string

↳ varchar

varchar(x)

0-255

initial allocation

avg length

→ run length encoding

abc → 3

↓

1/2 bytes for run length

→ 3abc

Text

- TINYTEXT 255 B
- MEDIUM 16 MB
- TEXT 64 KB
- LONGTEXT 4 GB

TEXT → Cannot be indexed

→ Full text index

---

int, varchar ~~TEXT~~

{Full Text → Text}