

① Data types

- String ✓
- Numeric

② Schema design

- What data goes in which table?

- Cardinality

- Case Study

Data types

CREATE

name:

);

TABLE 'users' (
String → Varchar

①

String

②

Numeric

- ③ Boolean
 - ④ Date (Temporal) Time
 - ⑤ BLOB's / CLOB
 - ⑥ JSON
 - ⑦ Enums
-

String

├ Char

└ string

- STRING

① char(x) — x — maximum

↳ fixed length string

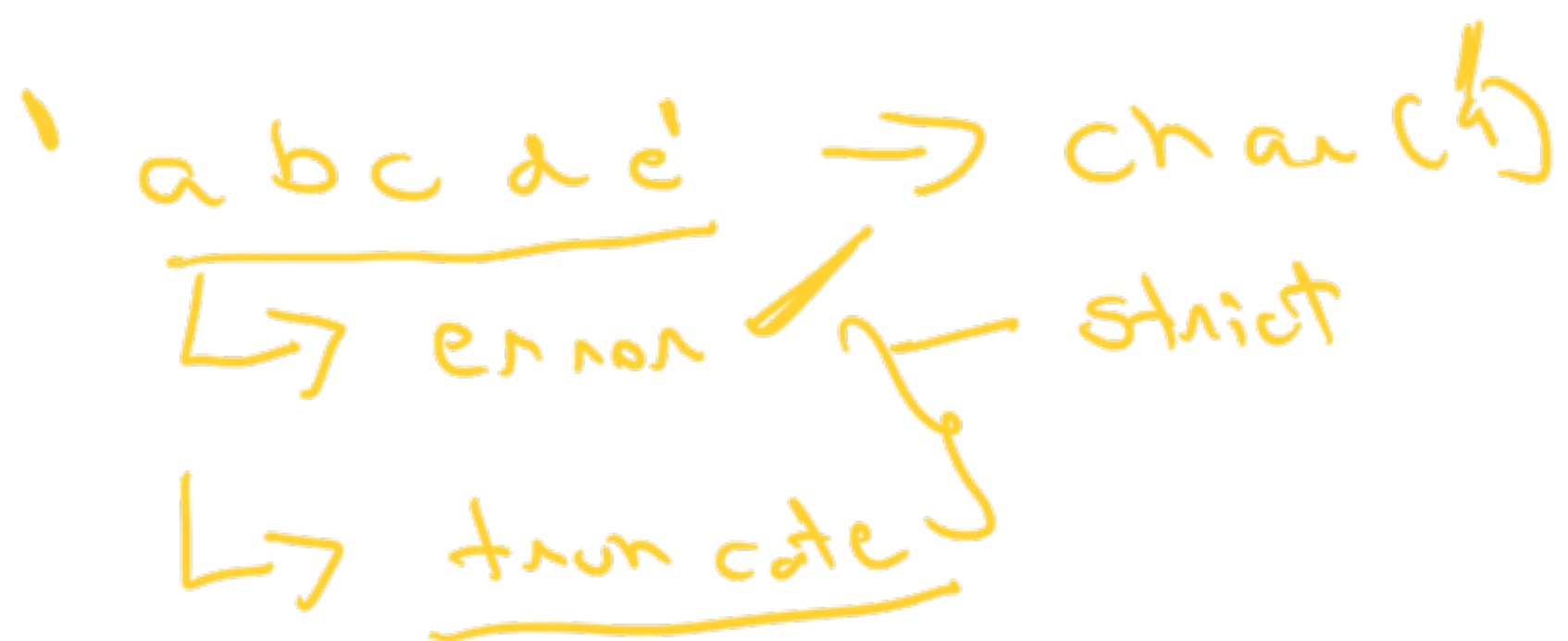
char('a') — ④

'abcd'

'a' → char('a')

' '

a _ _ _



x - [0 - 255]

- My Spl - strict mode
- strongly typed

What can we store in char?

→ Phone number codes

→ Country code - IN / UK

→ Gender - M F

→ padding $\sim Lx$

②

varchar(x)

$x \rightarrow 0$ to 65,535

$x \rightarrow$ initial allotment
 \rightarrow avg. length

Variable → run length encoding

abc - abc
- 3abc
↑

1/2 bytes

↓
length of the string



③ Text

- variable length

- lot of use cases

→ TINY TEXT 0 - 255, 5b3 = 255 B

→ MEDIUM TEXT = 64 KB

→ Long Text = 4 GB

Text → does not support indexes

Select * from user TEXT
where name = "Sakti"

String



- Char - fixed length - IN, UR
- varchar - variable length
- Text - more data

CHAR

TINYTEXT

MEDIUMTEXT

Full text indexed - TEXT

Schema design

↳ Structure

Schema → blue print of your database.

→ tables

→ fields

→ FK

→ PK

User — id

- name
- phone

Design

- entities / table
- data types
- visualization
 - class
 - ER

diagram

diagram

Schema
Design

① Requirements

② Schema design

③ Implement.

Case Study: ReScoler

① Find all entities



real world

or conceptual

objects of

interest

Student
user

Batch

How can we identify entities?

- Identifying all the nouns

A user will login

noun \rightarrow entity

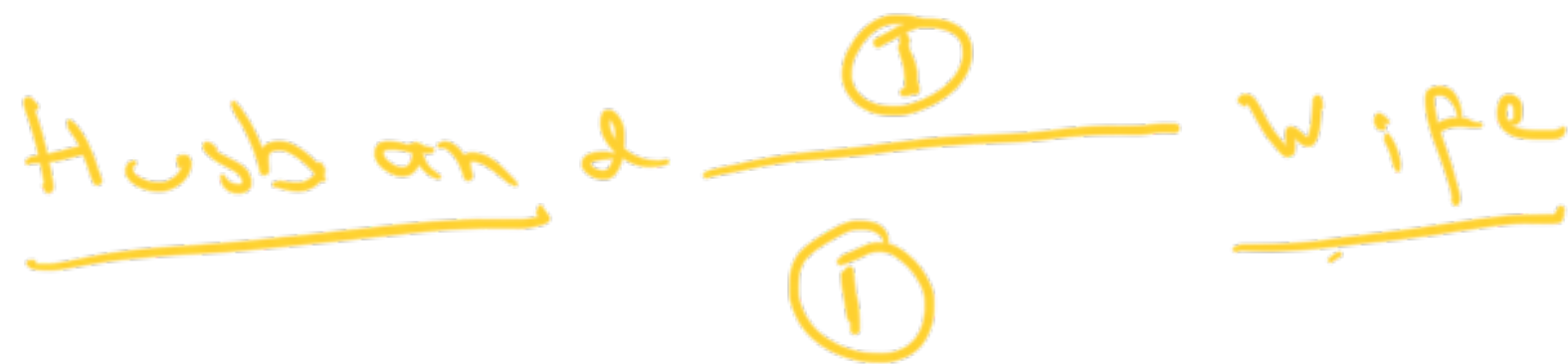
Cardinality

A — B

If there is a relation b/w A & B

Cardinality? How many A:

How many B



one to one \Rightarrow 1:1



How to establish cardinality





max (1, M)



1: Many relation





1: n



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

① A

② B

Student

Batch



• CU next batch

• previous batches



① m:n m

M:N



Cardinality

↳ where should a field be ^{attribute}



1a	
①	

1	1
2	1
3	2

③

↓
1 : M
or
M : 1

FK on the M side

③

③

M

(5) - (1)

Student	
id	batch_id
1	1
2	1

batch	
id	student_id
1	1
1	2

M : N

Student

id	batch_id
1	2
1	3

Batch

id	student_id
1	1
1	2

M : N

Separate + able



Mapping + able

1 - batch - prev - batch (PK)

Student - primary key

id	(Student_id batch_id)
1	1
2	2

Composite = (Student Id, batch Id)

P
id } uniqueness.

Student

~~(10)~~ name | age | uni | ^{name} batch_id

(1) ~~(1)~~ Total | 100 | Shweta | 1

Total | 100 | Shweta | 2

10M

Relational → relations

→ FK

[1, 2, 3]

LONG TEXT
PRINT

6:14 6:20

10:50

AUTO_INCREMENT

Cardinality = Caveat / Side problems

Schematische Design

11:30

RLE - 3abc

→ 1:1 → FK on any side

→ 1:M or M:1 → FK on the M side

→ M:N

Student
id batch

Batch

duplicate

$id_1 = id_2$

1	1
1	2

--	--

Mapping table

duplication

id	sid	bid
1	1	1
2	1	2

(SID, BID)

Caveat #1

Null values



student	mentor_id
1	2

2		3
3		NULL}

Student_id	mo ck_interview_id
1	NULL
2	NULL
3	NULL

} }

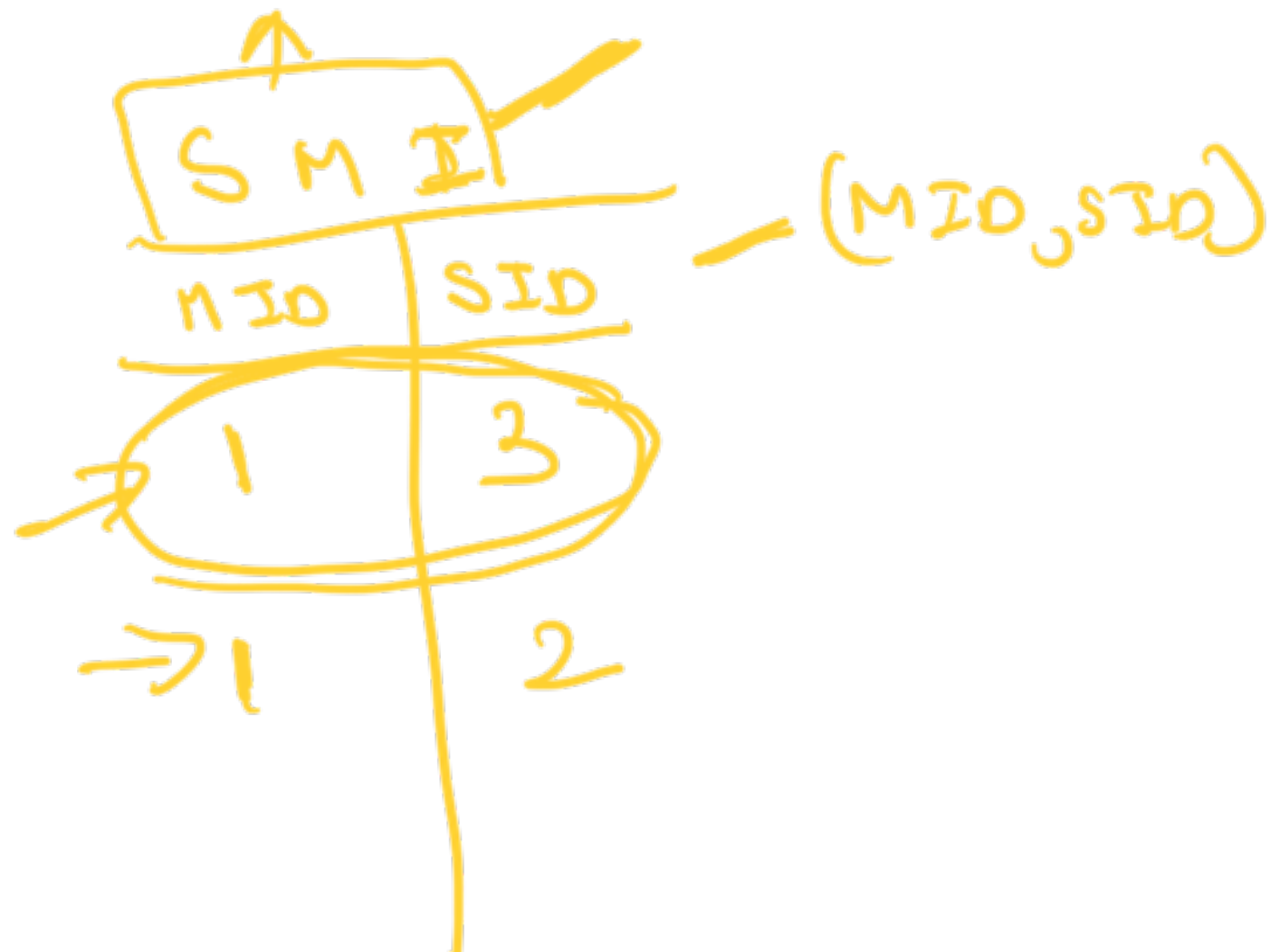
Sparsity table

We have / expect a lot of null values

it is better to use a mapping table

Student	
1	a
2	
3	

Student mockinterview





Join → Caching
→ SQL - Caching

phone number

SID | phone_number

(first name)

(last name)

Caveat #2

Attributes

slower

Student

id | batch_id |

start_date | end_date | pause_date

Attributes for a relation

S to start - id
id | name

S b
bid | student | start to
end to

Data types

- String



char(x) Fixed

max length



vchar(x) - Variable

initial allocation

time



⇒ Case Study

→ Schema Design

→ Requirements ⇒ Entities

→ nouns → entities

⇒ Entities ⇒ Table

→ Identify relations

→ Cardinality

How many of Entity A
: How many of Entity B



placement of FK

1:1 → on any side

1:M → M side

m:1 ✓

m:n ✓ → mapping table

Caveats

→ Sparse → MT ✓

→ a lot of attributes on the relation → MP ✓

Assignment

→ Netflix

→

→ Identify entities

→ Schema design

SOL: → CRUD

CREATE TABLE students (

PRIMARY KEY (id, name)

✓

→
→ of sys.

- ① MySQL - DBMS
- ② MySQL WB - Client

① Req

- ② Ask of on req
- ③ Identify partitions
- ④ Relations

String

① char(x) \rightarrow max length

- fixed length string
- padded

char(n) 0-255

\hookrightarrow "abcd" (0-65,53)

\hookrightarrow "a..."

\hookrightarrow "..."

↳ "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





Text

↳	TINY TEXT	255 B
↳	MEDIUM	16 MB
↳	TEXT	64 KB
↳	LONGTEXT	4 GB



TEXT \rightarrow Cannot be indexed

\rightarrow Full text in Arc

int, branch, ~~TEXT~~
{ Full Text \rightarrow Text }