SQL - III - Data types and normalisation

Agenda

1. Data types
   - String
   - Numeric
2. Data Normalisation
   - Anomalies
   - Normal forms

---

1. Data types

   1. String

2. numeric
3. boolean
4. Floating ⅋
5. Date
6. BLOB

---

1. String → Length (Max)

CHAR(X)

- Padded string
- fixed length

CHAR(4)

→ "abcd"

abc

→ 4

→ 4

100 - 10 90

CHAR(4)

↳ "a"

↳ "a - - - "

↳ Length - 4

Use cases

- Grade - A, B, C, D

CHAR(1)

→ Gender

→ Pincode

→ country code

→ PAN ADHAR

0 - 255

Varchar

- Variable character

- VARCHAR (4) → 0 - 65,535

4 - @

CHAR - ~

✓CHAR - @

VAR CHAR (255)

abc → 3abc

abcde → 5abcde

100 ... → 100

Run length

VCHAR (10)

5 abcde

1 byte

* Text

- TINY → TEXT
- MEDIUM
- LONGTEXT

Tiny - 256B
TEXT - 64 KB
MEDIUM - 16 MB

LONG - 4GB

Text cannot be indexed

- costly bad performance

AWS

S3 - Filestore

Numeric

	Storage	
TINYINT	1B	127
SMALL INT	2B	
MEDIUM INT	3B	
INT	4B	

Age - 100 127

└ [binary] | 8b |

1 b → [255]

→ -127 - 126

2 b → 32K

---

1 b — 8 bits

(□□□□□□□□

$2^8 = 256$

0 - 255

/2

-1

-2 - 127 — 128

Floating point

(1) DECIMAL (P, S)

Precision → length

Scale → decimal points

302.04

p = 5

s = 2

DECIMAL (5, 2)

100.98 / 101.92

302.505 | 302.505|76

Truncate

302.505

302.505 D(6,3)

p → length → 6

s → decimal → 3

D(6,3)

302.500

(2) Float

- approximate

IEEE 754

Epsilon → buffer

$1.9 - 0.9 \pm \delta == 1.0 \pm \delta$

$0.0001 - 0.001$

$1.09 - 0.09 \neq 1.0$

$1.89 \quad 0.91 \pm \delta = 1.0 \pm \delta T$

BREAK

6:00 - 6:05

10:35

Data Normalisation

Anomalies - Error

- Outlier

Student - Batch

Student Batch

ID	NAME	E	BID	BN
1	NULL	NULL	1	April 23

Insertion Anomaly

2 | June 25

Deletion Anomaly

ID	Name	BID	BName
1	Sherlock	1	S
2	Rocky	1	S
3	Kitush	2	D

Student

1	S	1
2	M	1
3	K	2

Batch

1	S
2	D

Updation Anomaly

SID	Name	BID
1	A	1
2	B	1
3	C	2

Batch A

BID	BNAME
1	Batch 1
2	AB

C B

Anomalies

X ① Insertion – We cannot create a batch w/o students

X ② Deletion – when we delete a student then we also delete batch

X ③

③ Updation - Due to human error,

→ Miss updating all rows

→ Inconsistency

Normalisation → Avoid anomalies

Boyce - Codd

INF

→ Only atomic values

→ single valued

→ No lists, collections

Students

ID	Name	Phone numbers
1	A	[+91 ..., +44 ...]
2	B	+91 ....
3	C	+44...

1. Add a new row ✗
2. Add a new column ✗

③ Mapping table

SID	SNAME	PHONE_01	PHONE_02
1	A	+91	+44
2			
3			

① Sparse column

2. How to decide # of cols
3. Querying is not easy

Student

SID	NAME
1	A

Phone

SID	PHONE
1	+91
1	+44

1. ID
2. Composite (ID + Phone)

③ iPhone

2NF ✓

① 1NF ✓

② No partial dependencies

---

an attribute should not depend on part of the PK

2
Ax
B
PK -
ID , BID
ID , BID
A
ID , BID
PSP
BID
B_NAME
Partial Dep.

Sol - Mapping table

Student

ID | NAME | DOB

Student_batches

SID | BID

BATCH

ID | NAME

MySQL

Client

1. Workbench
2. TablePlus

API
↳ CRUD