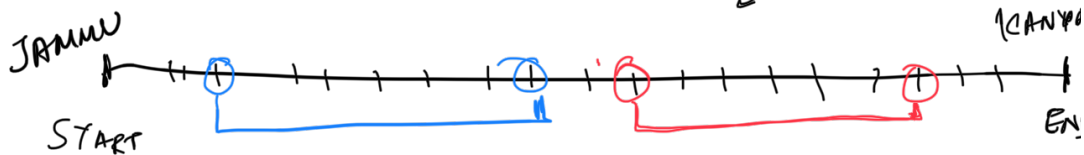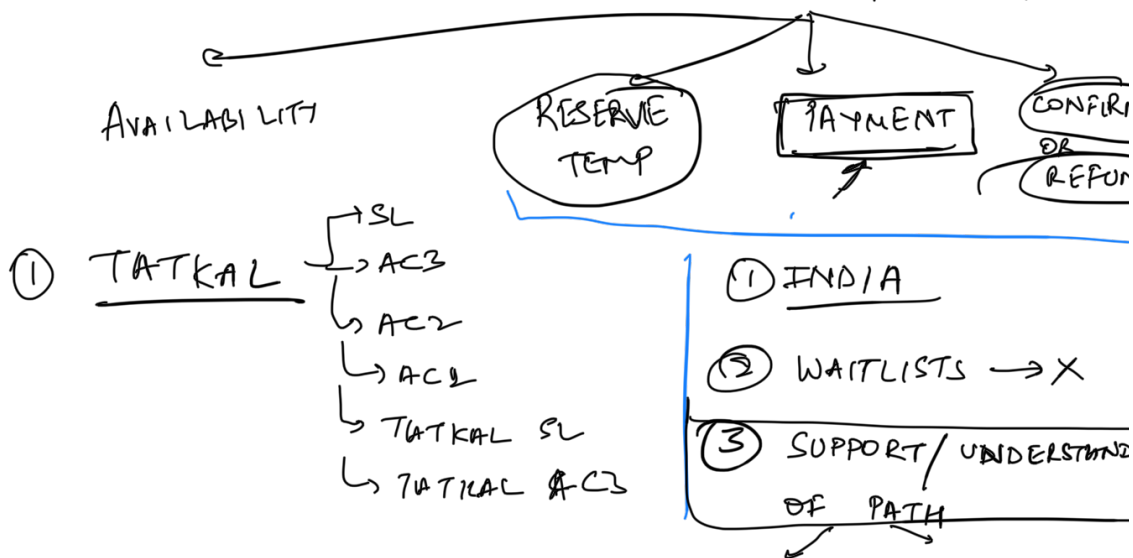# DESIGN IRCTC

**STEP 1:** SOURCE — DESTINATION, DATE

① Give me the list of trains along with expected start and end time

**STEP 2:** GIVEN TRAIN_ID, DATE, SOURCE, DEST, CLASS

⤷ AVAILABILITY STATUS.

**STEP 3:** BOOK A TICKET ~~BY~~

⤷ TRAIN_ID, SOURCE, DES

DATE, # of seats

↓

PASSENGER DETA



AVAILABILITY    RESERVE TEMP    PAYMENT    CONFIR

OR

REFUN

① TATKAL ⟶ SL

⟶ AC3

⟶ AC2

⟶ AC1

⟶ TATKAL SL

⟶ TATKAL AC3

① INDIA

② WAITLISTS ⟶ X

③ SUPPORT / UNDERSTAND OF PATH

JAMMU                    (CANY

START                    EN

---

ESTIMATION OF SCALE

4000 – 6000 trains per day

↓

20 BOGIES

5000 * 3200

≈ 15 million se...
↓
20% extra booki...
for waitlist

= 18 million bookings

20 * 8 = 160 seat...

10 calls write queries

Write QPS → 18M * 10 → 180 Milli...
queries

9am ——— 10an —— 11an = 12pm — 1pm · — — — · —

↑
15th    $\frac{180}{5}$ = 36 Million queries

20% of
bookings ar
Tatkal

$\frac{36M}{60 \times 60}$ → 10K queries per second

18 M bookings per day

| Station_id | Source | dest se | Train_id | Coach-no | Seat_No | Passenger-Na | Age | II |
|---|---|---|---|---|---|---|---|---|
| 12 b | | | 4 bytes | 4 byte | 4 bytes | 128 bytes | 4 byt | LBy |

18 M * 200 bytes = 3.6 GB

3 months of data / 90 days.

fit in
one
mach

≈ 90 * 3.6 ≈ < 350 GB

High consistency / High concurrency

esign goals

S1        High AV

List Train Search          Booking + Availability          Payment

S2

J S3

Train DB
source, dest, days

Confirmed

① High consistency
   ↳ 2x booking ✗

↷ Razorpay
   Async

② High concurrency. (10k writes per second)

---
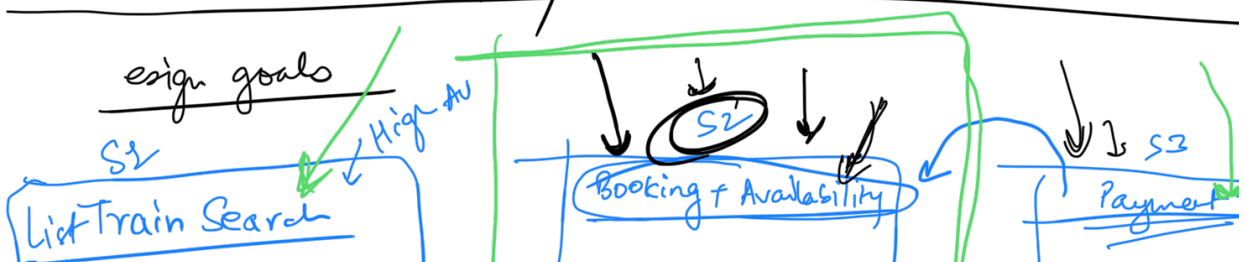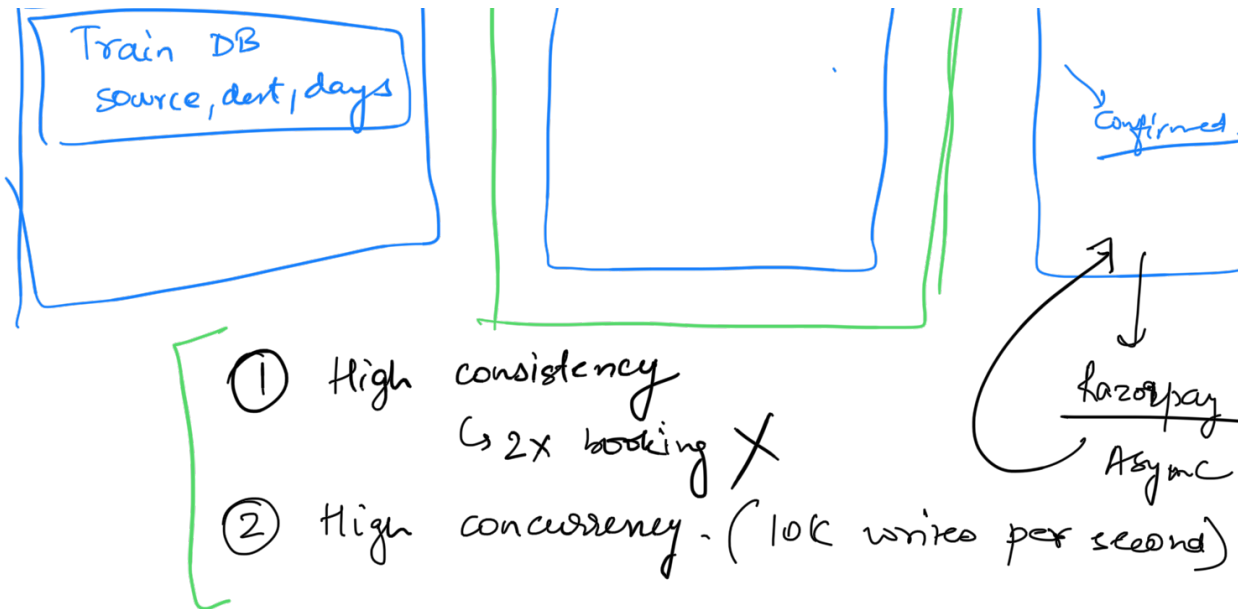
Booking + availability service

API
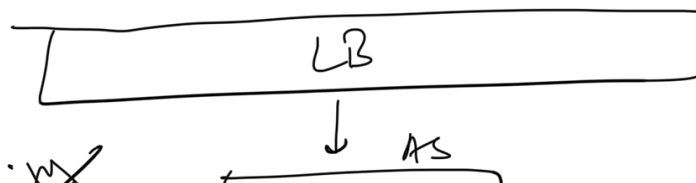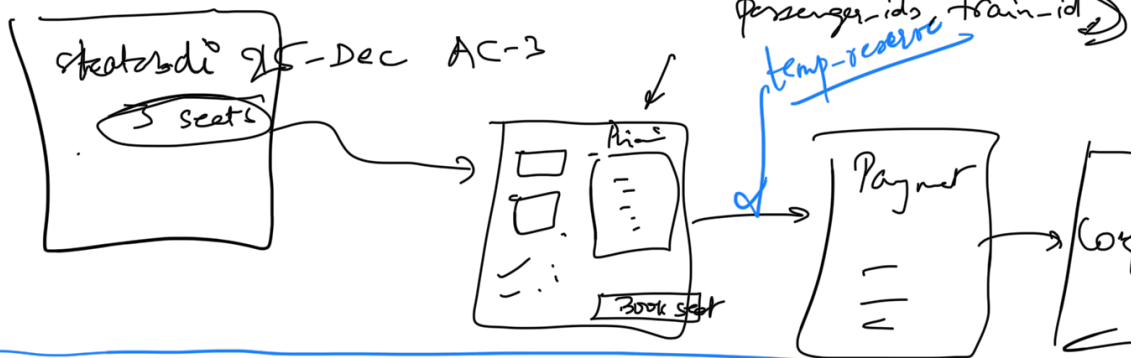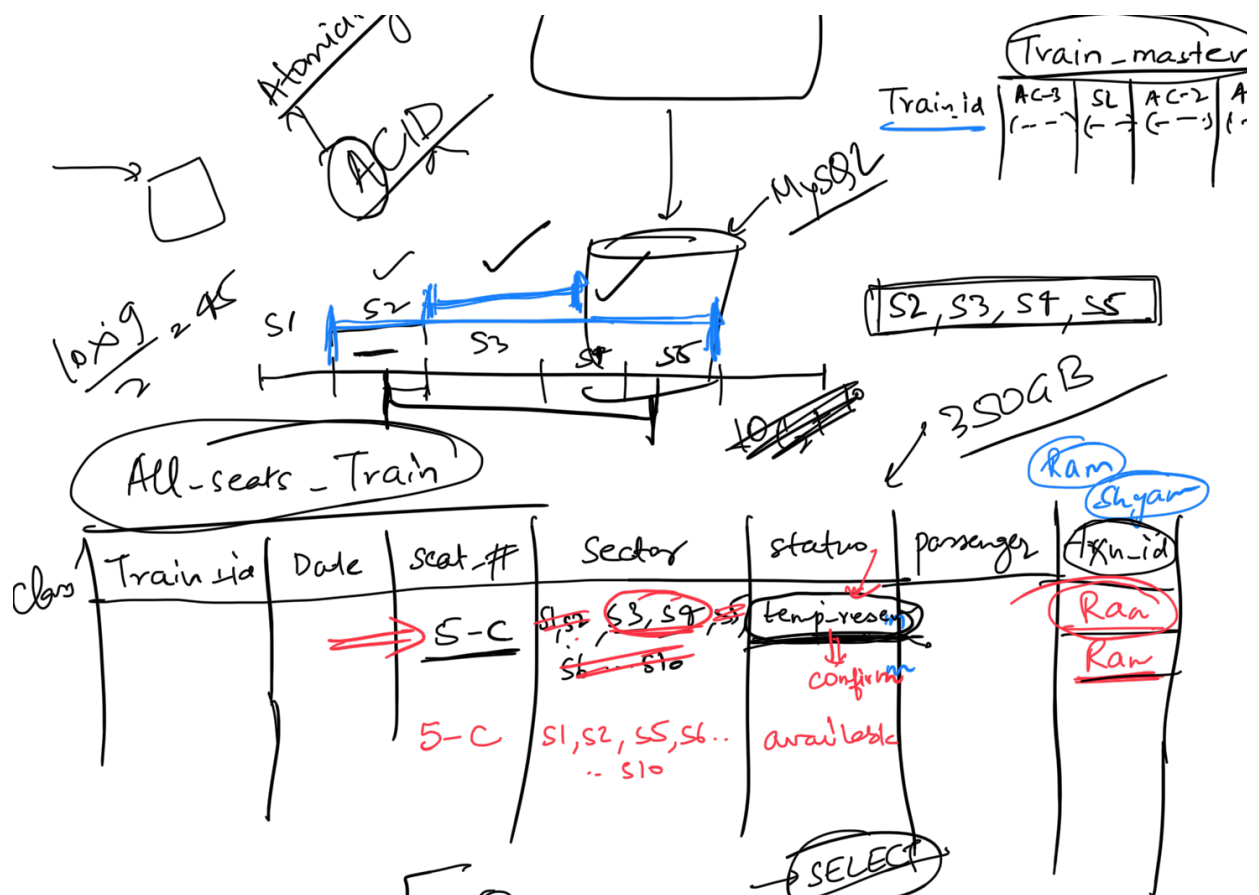- check Availability (train_id, dat
                      src, dest, seat_class)
→ temp-res
- bookSeats( train-id, date,
            src, dest, seat-class,
            # of seats, list of passenger
                              details)

- cancelSeats( - . - )

→ confirm Seats ( # transaction-id,
                  passenger-ids, train-id)
                  temp-reserve

seatsdi 25-Dec AC-3
(5 seats) → [booking] → temp-reserve → Payment → Cor
                         Book seat

---

LB

. WX        ↓ AS        ↓

Atomic...

ACID

Train_master

Train_id | AC-3 | SL | AC-2 | A

log 2 45
2

S1  S2  S3  S4  S5

MySQL

S2, S3, S4, S5

350GB

Ram
Shyam

**All_seats_Train**

| Class | Train_id | Date | Seat_# | Sector | Status | Passenger | Txn_id |
|-------|----------|------|--------|--------|--------|-----------|--------|
| | | | 5-C | S1, S3, S4, S5 / S6 - S10 | temp-reserv | | Ram / Ram |
| | | | 5-C | S1, S2, S5, S6... -- S10 | available | | |

confirm

SELECT

Atomic ← [
① Check tn., de, class, which seat is ave

② block the seat → UPDATE
]
↙ upd

O ← [
UPDATE  TRAINS  SET
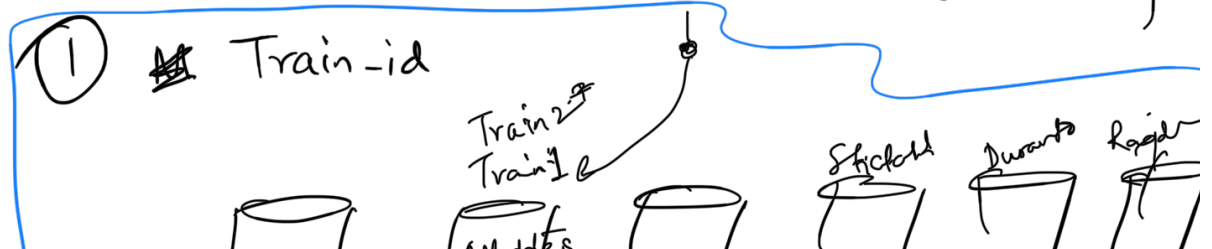status = "temp-reserve",
txn-id = * my_txn_id
ts = NOW()

WHERE

(status = "available" OR
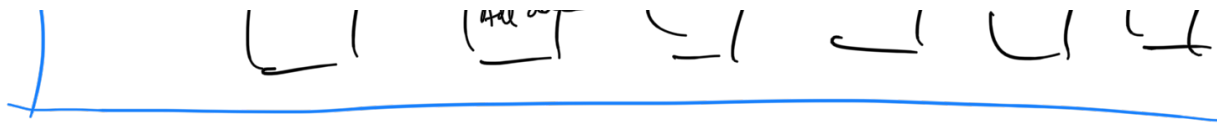(status = "temp-reserve" AND ts < NOW))
AND train_id = XYZ
AND: class = abcd
AND date = "28-12-2021")
AND sector LIKE '%S3%' AND sector L
'%S
LIMIT # of tickets
]

① # Train-id

Train2
Train1

Stickets  Duranto  Rajd

(2) Date
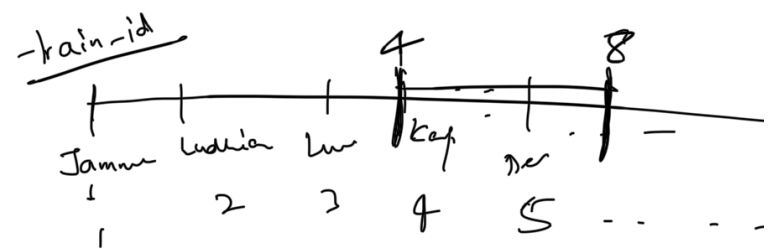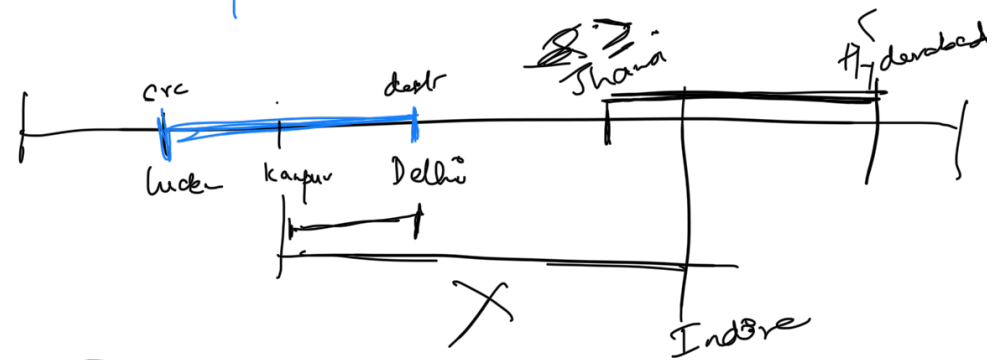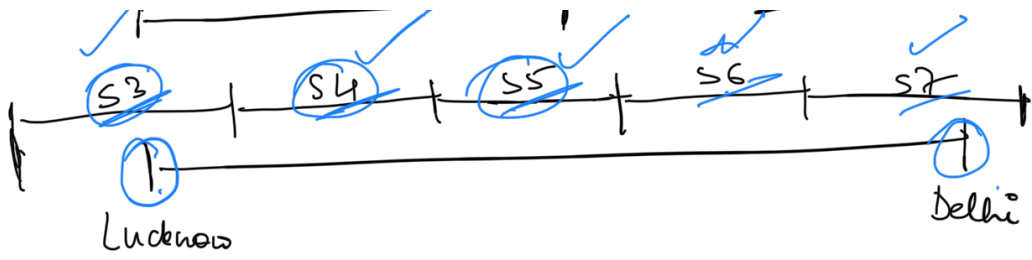T+90        T89                                    Tf1

(1) Shard based on trains to
    reduce load per DB

(2) Use atomicity of SQL Databases to
    prevent 2x booking.

                              Jhansi              Hyderabad
    src              dest
Luck    Kanpur   Delhi
                                                  Indore

-train-id              4           8

Jammu  Ludhia  Lur  Kap    Der
  1      2      3    4       5  - - - - -

train id | date | Source | dest

Lko                    Del
                 Del          Kolketa

S3  S4  S5  S6  S7

Lucknow          Delhi

| Train Id | Date | stop id | seat # | class | status |
|----------|------|---------|--------|-------|--------|
|          |      | LKO     | 5-c    |       | av     |
|          |      | Ka      | 5-c    |       | a      |
|          |      | :       |        |       |        |

S7        S6
LKO  d    del

sector line    "....⊙⊙⊙⊙...."
                ↑
               0/1

%  XYL %

Improvement

BOM      CHENNAI      SINGAPORE

Indig

                    Boit

Vistara                    United

X

Y