

SCHEMA DESIGN - 3

- STEPS
- THINGS TO KEEP IN MIND
- NETFLIX CASE STUDY.

Steps

- 1 [Nouns → entities] ✓
- 2 [Relationship / cardinality between entities] ✓
- 3 [Primary Keys / Keys unique] ✓
- 4 [Index] ✓
- 5 [Enums] ✓

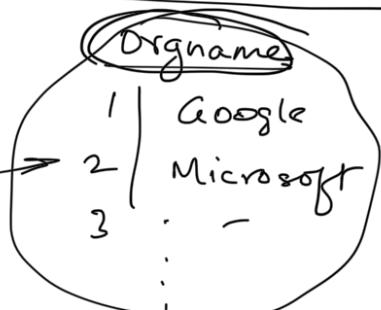
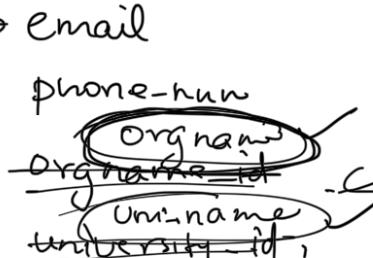
① Entity vs Attribute

② Too much normalisations.]

[Google]

Users → email

COST



enum

User-state → active / inactive

0 / 1

From user → red / blue arrows. ... →

User	
if	level ... state
	O
	L

User-state-lookup	
(0)	inactive
(1)	active

Tree → unique values of column where index is created
 value in the tree is list of pointers to matching rows in table

key → column position which index is created

value → all matching rows in the table

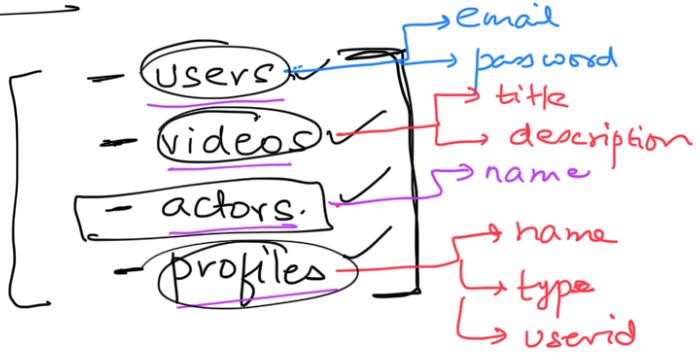
SELECT name
 FROM users
 WHERE (Orgname = "Google")



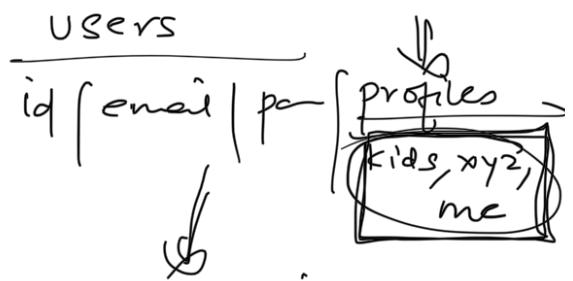
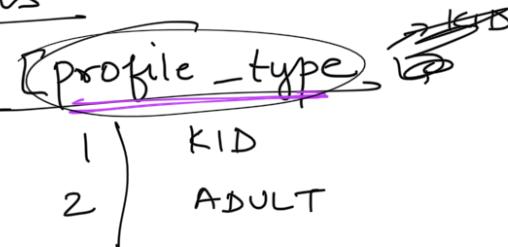
Orgname LIKE "%Google%" → Full text search
 Score > 500 → Regular exp
 100 → 100

without archive

Step 1: find the nouns.



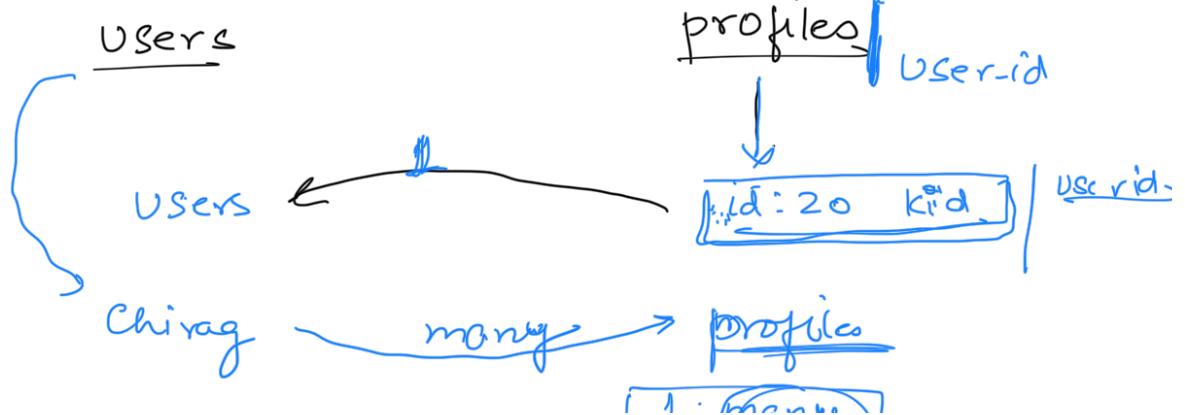
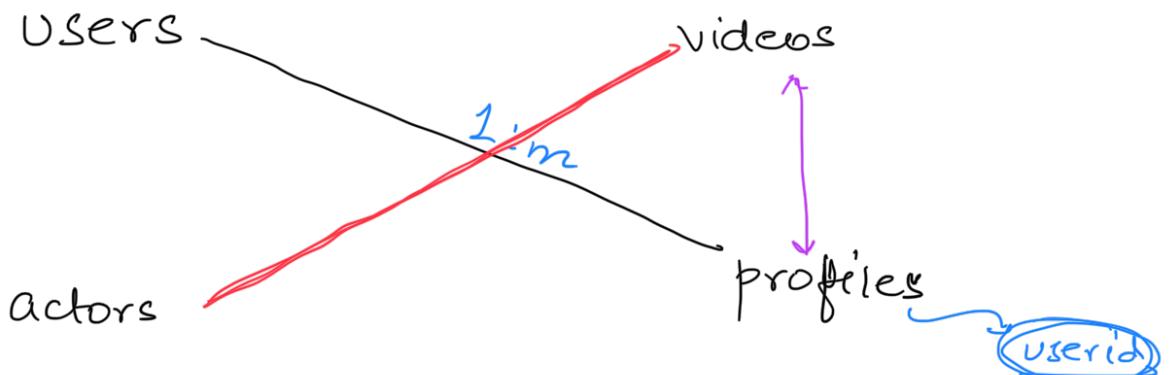
Enums



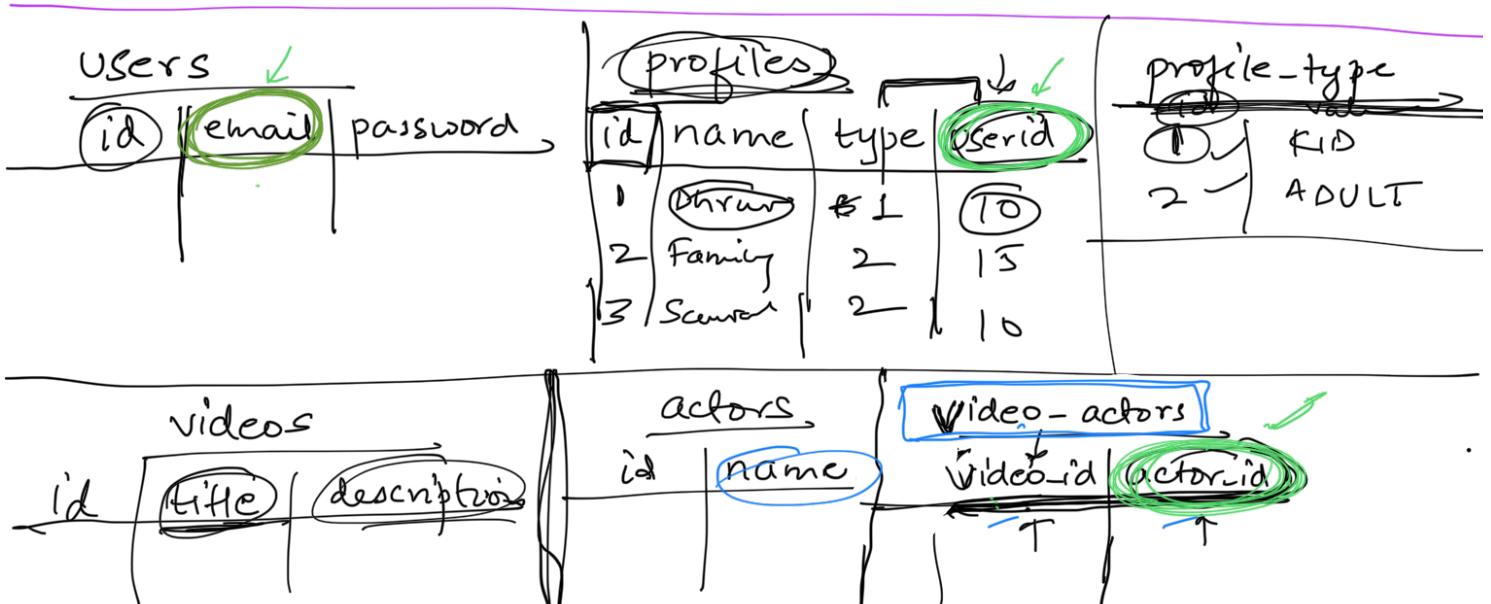
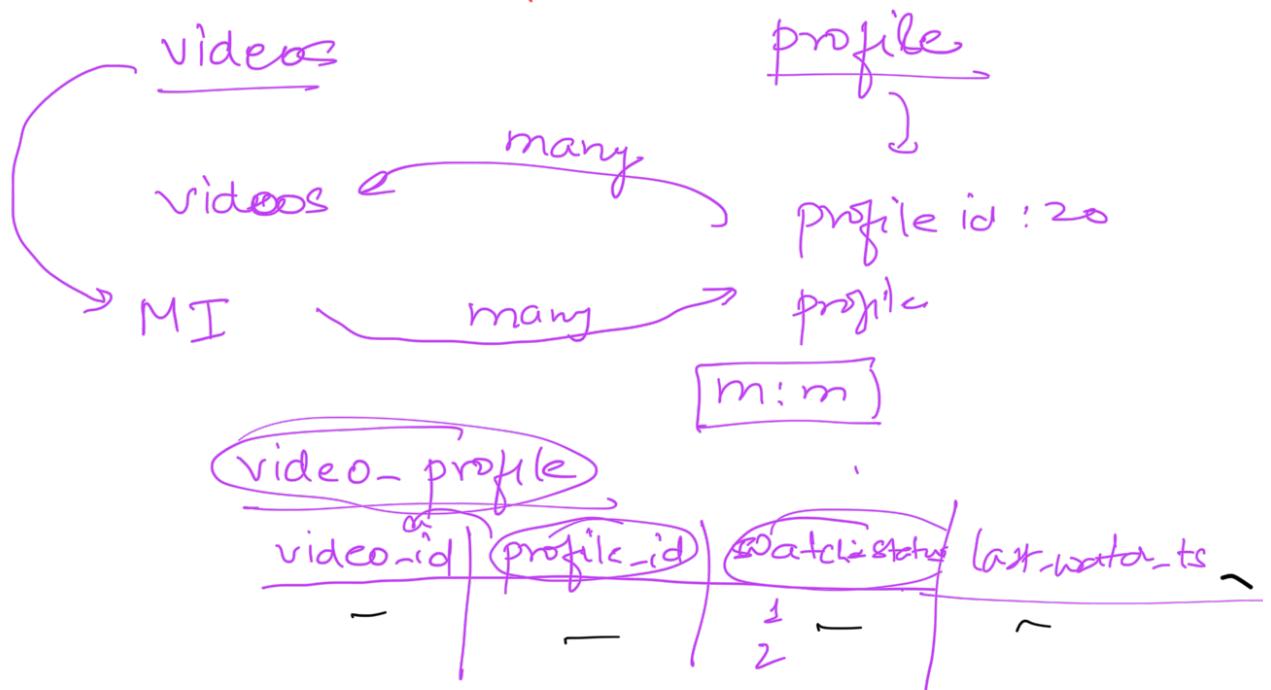
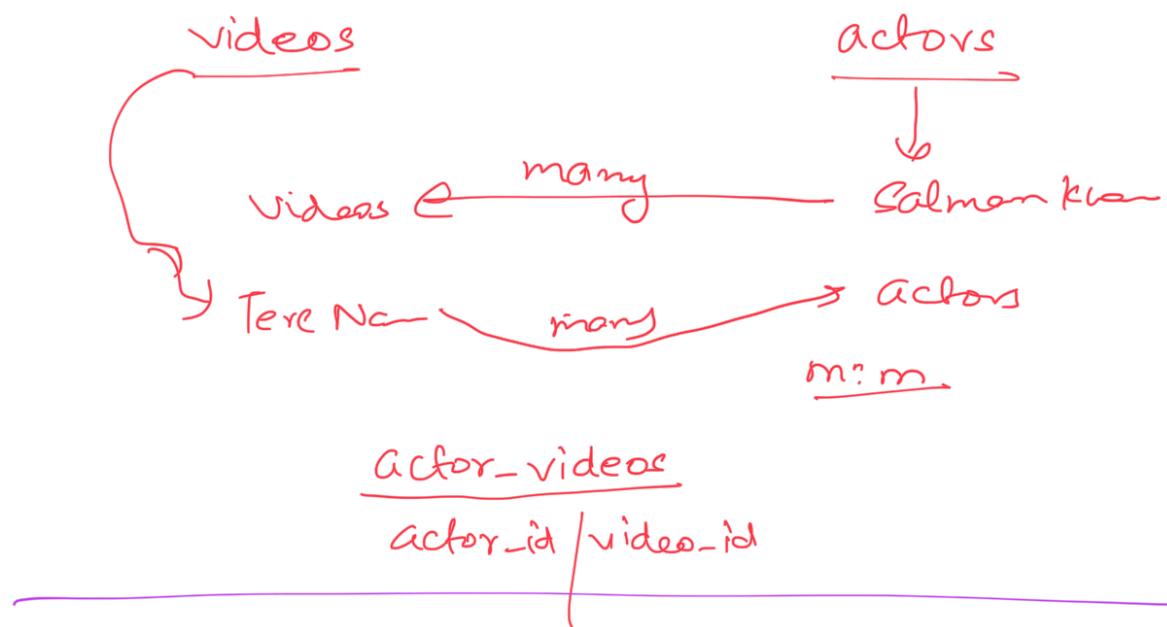
Watched-video-status

1	COMPLETED
2	IN PROGRESS

User-profiles



L - "Many"





① Log In

→ email → password

SELECT password FROM users WHERE email = %

②

userid

↳ [profile_id, profile_name] profile type.

SELECT id, profile_name, type,

FROM profiles

WHERE userid = <--->

WHERE
JOIN

③

Recently played videos

userid, profile_id

List of all videos

SELECT title, desc, video-id

FROM video-profile

WHERE

profile_id = <--->

AND watch-status = 2

ORDER BY last-watched-ts DESC

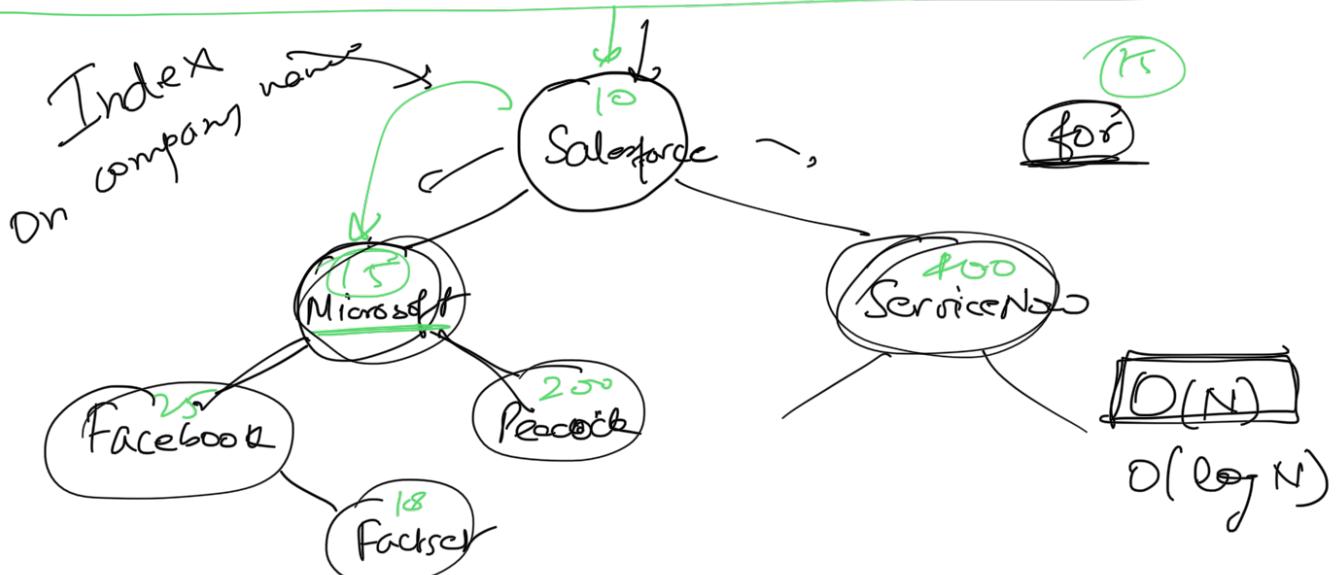
LIMIT 10

<u>profile-id</u>	<u>watch</u>
1	1]
2	1 ↗
3	1
4	1
5	1
6	2

↳ (profile-id,
watch-status)

④ video-id → list of actors.

SELECT
(name)
 FROM video-actors va
 JOIN actors a
 ON va.actor_id = a.id
 AND va.video_id = <...>
 Salesforce, 10



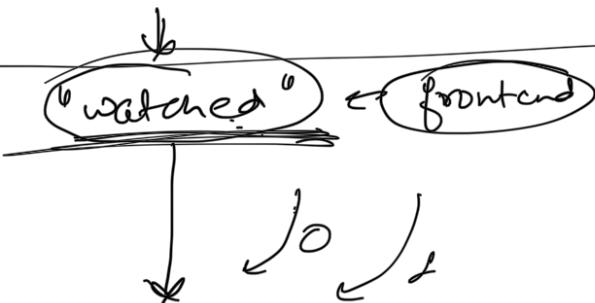
$$J(a, b) \leftarrow \boxed{b = 10}$$

mysql -u root -p '...'



Subqueries

- Pros → readability
- Pros → easy to write
- Cons → cannot use index on result from subquery



API → getMovies(~~profile_id~~, ~~is-watched~~, num_entries, offset)

Java backend → bugs

watched-status = (is-watched? "COMPLETED": "IN PROCESS")

SELECT

FROM videos_profile VP

| 0 |
JDN watch-sta us
ON vp.watchsta_id
= ws.id

AND ws.status =
Good ->

