Lecture : Schema Design -II

Agenda
- Recap: Tables, Attributes, Relations, Primary Keys & Foreign Keys

- Relationships
    - 1:1 id of any side can go on other side
    - M:1 id of one side goes on many side (student - batch)
    - M:m mapping table

- Mapping Table / Separate Table

    i) m : m cardinatlity (orders & products)
    ii) Sparse Relation (students shifting batches)

- How to decide upon indexes?

TODO NETFLIX [9.30 PM]
Case Study:

Design Database Schema for a system like Netflix with following Use Cases.
Use Cases

1. **Netflix has users.**
2. **Every user has an email and a password.**
3. **Users can create profiles to have separate independent environments.**
4. **Each profile has a name and a type. Type can be KID or ADULT.**
5. **There are multiple videos on netflix.**
6. **For each video, there will be a title, description and a cast.**
7. **A cast is a list of actors who were a part of the video. For each actor we need to know their name and list of videos they were a part of.**
8. **For every video, for any profile who watched that video, we need to know the status (COMPLETED/ IN PROGRESS).**
9. **For every profile for whom a video is in progress, we want to know their last watch timestamp.**

1. Create Tables (Nouns)
    - Users (**UserId**,Email, Password)

        Primary Key: User Id

        No foreign key

        No additional index is needed

    - Profiles (**ProfileId**, **UserId**, ProfileName, ProfileTypeId)

        ProfileId - Primary Key

        UserId - Foreign Key

        Indexing : on **UserId** for the below use-case

        Possible Use case -> A user logins in, he wants to see all the profiles that belong to the user.

        ```
        SELECT *
        FROM profiles
        WHERE UserId = 5
        ```

    -ProfileTypes (ProfileTypeId, ProfileTypeValue -> KID or ADULT)

        PrimaryKey: ProfileTypeId

        No foreign key

        No index is needed

    - Videos (**Video_Id**, Title, Description)

        Videos : cast = m : m

        PrimaryKey : Video_Id

        Foreign Key : No

Index : Title column

- Actors (Actor_id, Name)
- Videos_Actors (Video_Id, Actor_Id)
    - Primary Key: Video_id + Actor_id
    - Foreign Key: Video_id, Actor_id
    - Index: Video_id + Actor_id , can have an additional index based upon “Actor_id”

Query Find actors who acted in video “DDLJ”.
SELECT a.name
FROM videos v
JOIN videos_actors va

USING (video_id)
JOIN actors a

USING (actor_id)
WHERE v.name = “DDLJ”

Find all films acted by sharukh khan or actor_id = 5;

- Profiles_Videos [to store the videos watched or are being watch by every profile] [**profile_id, video_id**, timestamp, watch_status_type]
- Primary Key: profile_id + video_id
- Foreign Keys: profile_id, video_id , watch_status_type
- Indexes: profile_id for the user, video_id for the analytics platform
- Upon login, out of all shows you want to display of shows that you were watching

-
    - WatchStatus

1	IN PROGRESS
2	Completed

Profile Table

Me - Prateek	1
Father - ABC	1
Son - XYZ	2

Profile Types

1	ADULT
2	KID

-