

**Dimensional Modeling FOR Pay Per View System**

**(Team Members)**

**Vignesh Srinivasan**

**Naitik Shukla**

**Shobhit Jaipurkar**

**Navneeth Goswami**

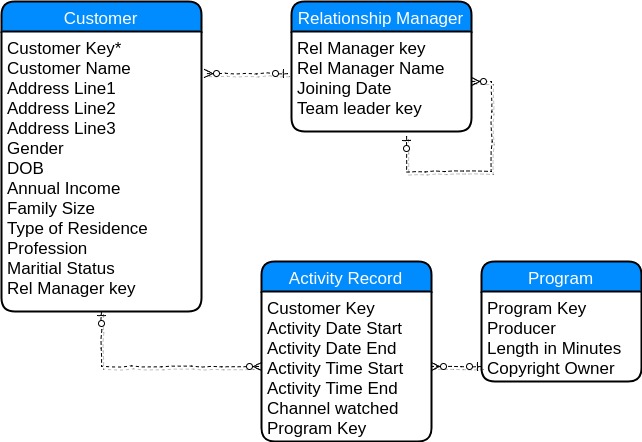
**Objective:**

We have to build a Data warehousing ( Pay Per View Model ) for MCTV (a cable TV operator in singapore media industry)

**Process for Modeling (Steps for changing Operational Model to Dimension Model ):**

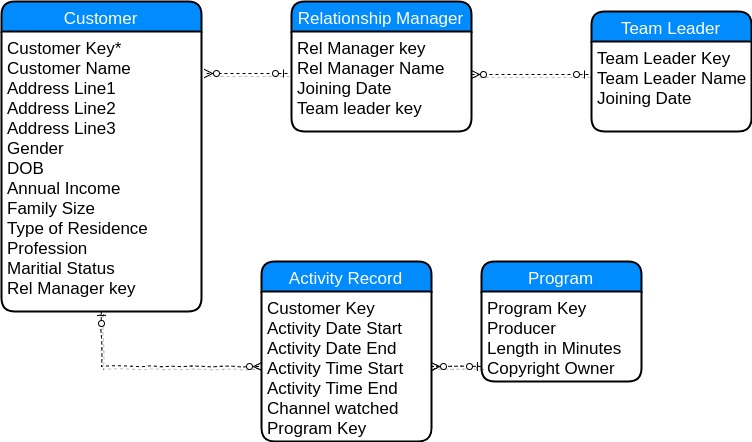
Removing attributes related to Operational Data ( Since Operational systems attributes are updated real time) Considering this we have selected some of the attributes like.

|  |  |
| --- | --- |
| Entity Name | Operational attributes |
| Customer | Last Maintenance Date , Don’t Bill This Month, Last updated workstation ID,Creation Workstation ID |
| Relationship Manager | Creation Workstation ID,Last updated Workstation ID, Last Maintenance Date |
| Activity Record | None (Since it's an transaction data we need to keep all the fields) |
| Program | Creation Workstation ID,Last Updated Workstation ID,Last Maintenance Date |

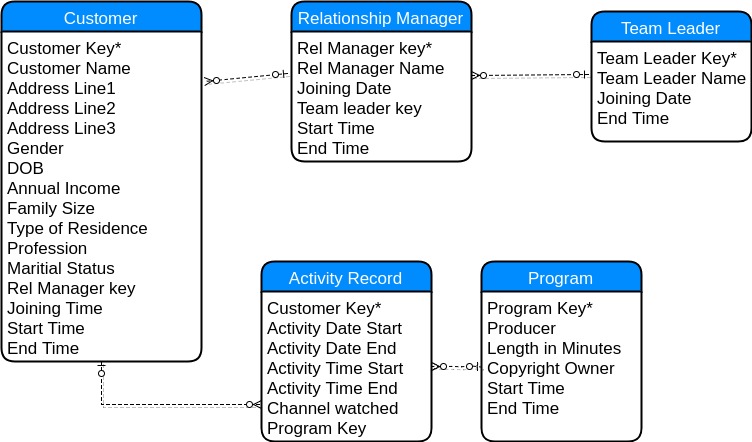


Step 2. Since all the relationship are based on a point of time , If the Team Leader can change under Relationship Manager (data might get lost as history not stored) So we can change this to artifact relationship which is correct for a snapshot of time.

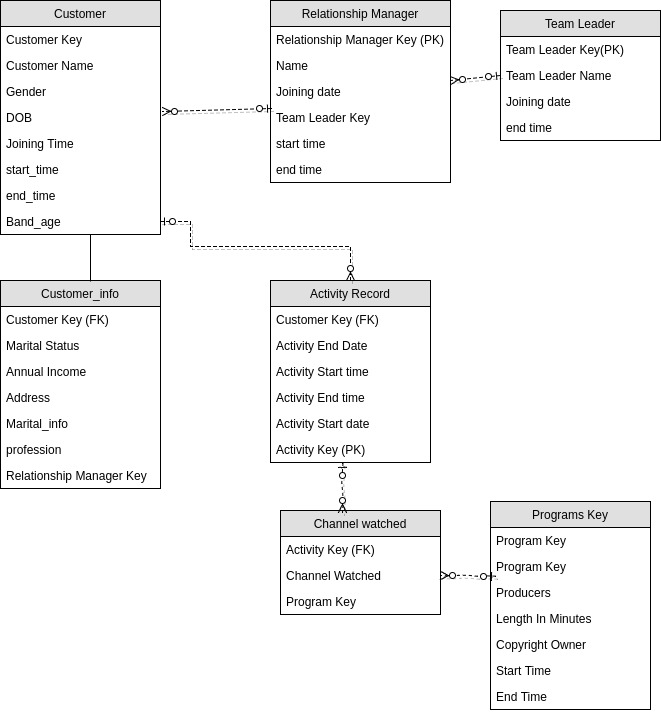
|  |  |  |
| --- | --- | --- |
| Artifact Inserted | Relationship | Attributes |
| Team Leader | Team Leaders -> Relationship Manager | Team Leader Key,Team Leader Name, Joining Date |



Step 3. Since we need to keep the record of changes for each entity , we can have time element at entity level



Step 5. We have separated attributes based on the frequency of updation



**Final Dimensional Modeling** -

We have selected snowflake schema for our Dimensional Modeling and Activities as fact table ( Since it's an transaction data) and as derived variable we have added **( Total Views , Total Programs Watched ,Total Cost)**

