



Project Requirements Document



Team Number: 1

Team Members:

- Bhav Beri (2021111013)
 - Harshit Aggarwal (2021111015)
 - Sreyas S (2021111016)
-

Introduction:

The project is a mini-world that represents, stores, retrieves, and studies the data of a standalone idea called **IIITVR (IIIT Virtual Relationship - We make Virtual Reality to Relationship)**. The design and structure of the database would provide an efficient data system for reporting and retrieving data uploaded and linked to the **IIITVR** website.

IIITVR is a virtual relationship provider. It helps people maintain a virtual relationship like in the metaverse. The users can opt for their preferences based on which a virtual environment and people are created to accompany the user. The platform lets users chat with other users/avatars in the metaverse and post pictures online. This mini-world fulfils the line - "Love is better off as virtual. You should feel it from far. Rather not touch it. And it will always be there for you."

Purpose:

With over 30 Million users for other similar platforms, handling data becomes vital. Handling massive data load sheerly using text or simple CSV files would slow handling, updations, and retrievals. This database is designed to reduce the complexity of the stored data and efficiently retrieve data stored in the database for a better user experience and faster querying time.

Users:

The maintainers of **IIITVR** will be sophisticated users. They will be actively involved in perfecting the database according to their needs. Other employees will be casual users of the database. The users of the service will be naive users of the database.

Applications:

The maintainers can use the database to remove profiles of fake identities or profiles that have violated regulations. The users will indirectly use the database to view other people's profiles. It can be used for storing, retrieving, processing, and understanding the information uploaded to IIITVR.

Database Requirements:

- Assumptions -
 - A user selects his partner in Meta-Reality from the current user's database only (As it's big enough to provide all options)
 - Users can't use Meta-Reality products without choosing plans or filling in their preferences.
 - Chats, once sent, cannot be deleted.
 - An issue is assigned to only one employee only
- Strong entity types-
 - Employees
 - User
 - Chats
 - Plans
 - Issues
- Weak entity types-
 - Photos
 - Preferences
 - Meta Reality
- Entity Types & Attributes -
 - Employees -
 - Employee ID - int
 - Employee Name - varchar
 - Department - varchar
 - User -
 - User ID - int
 - Aadhar Number - int (Candidate Key)
 - Name - varchar
 - Gender - varchar
 - Age - int (derived)
 - Date of birth - time & date
 - Joining date - date
 - Plan ID - int (Foreign Key)
 - Chats -
 - Sender - varchar
 - Receiver - varchar
 - Timestamp - time
 - (COMPOSITE PRIMARY KEY - Sender, Receiver, Timestamp)
 - Content - text
 - Plans -
 - Plan ID - int
 - Cost - float
 - Duration - int
 - Features - text (multivalued)
 - Issues -
 - Issue ID - int

- Timestamp - time
 - Initializing user - varchar
- Photos -
 - Media data - BLOB
 - Media size - float
 - Entry_Location - varchar (Partial Key)
- Preferences -
 - Gender - varchar
 - Age - int
 - Language - varchar (multivalued)
 - Country - varchar (multivalued)
 - Description - text
- Meta reality -
 - Location - varchar (multivalued) (Partial Key)
 - Time - date & time
 - Climate - varchar
- Relationship Types-
 - Makes
 - Employees make plans for the companies.
 - Between Employee and plan
 - Several employees can make a plan, and an employee can make more than one plan.
 - Cardinality - M: N
 - Degree = 2 (Binary)
 - Participation Constraint - Employee(0, M); Plan(1, N)
 - Resolves
 - Employees resolve different issues (created by Users).
 - Between Employees and issues
 - An employee can solve several issues. Only one employee can solve an issue.
 - Cardinality - N:1
 - Degree = 2 (Binary)
 - Participation Constraint - Employee(0, M); Issue(1, 1)
 - Maintains
 - Employees maintain the virtual reality platform and product, making it always operational.
 - Between employee and meta-reality
 - Cardinality - M: N
 - Degree = 2 (Binary)
 - Participation Constraint - Employee (1, M); Meta reality (1, N)
 - Sends
 - User sends different chat messages, which are stored in Chats relation.
 - Between Chats and users
 - A user can send several chats, but a single user can send a chat.
 - Cardinality - 1:N
 - Degree = 2 (Binary)
 - Participation Constraint - User (0, M); Chats (1, 1)
 - Pays For
 - User pays for one of the plans provided by the company.
 - Between users and plans
 - A user can select only one plan, but a plan can be chosen by several users.
 - Cardinality - N:1
 - Degree = 2 (Binary)
 - Participation Constraint - User (0, 1); Plan(0, N)
 - Opens

- Users raise / open issue requests they face
 - Between user and issues
 - User opens several issues/problems/support, but an issue can be opened by a single user.
 - Cardinality - 1:N
 - Degree = 2 (Binary)
 - Participation Constraint - User (0, N); Issue (1, 1)
- Posted By (Identifying Relationship)
 - An image is posted by an user on platform.
 - Between user and photos
 - An image can be posted by a single user, but a single user can post several images
 - Cardinality - M:N
 - Degree = 2 (Binary)
 - Participation Constraint - User(0, M); Photos(1, N)
- Uses
 - An user uses a meta-reality product which works based on the user's preferences and the features provided are based on the plan selected by the user.
 - Between Users, meta-reality, preferences and plan
 - A user uses one product based on their specific preferences and selected plan, but several users can use a single meta-reality product.
 - Degree = 4 (4-ary)
 - Participation Constraint - Users(0,N); Meta-Reality(0,1); Preferences(1,1); Plan(0,1)
- Updates (Identifying Relationship)
 - User fills and updates his specific preferences, and these preferences are user-specific.
 - Between user and preferences
 - A user will have particular preferences, and a preference will be particular for a user.
 - Cardinality - 1:1
 - Degree = 2 (Binary)
 - Participation Constraint - User(1,1); Preferences(1,1)
- Support Priority
 - Support Priority for a particular issue depends on the plan by a particular user.
 - Between plan, issue and user
 - Degree = 3 (Ternary)
 - Participation Constraint - Plans(0,1); Issue(1,1); User(1,1)
- Trains
 - Meta-Reality AI trains its dataset based on the photos posted by users (non-specific) to generate real-life replicas of different people for use with other people.
 - Between Meta reality and photos
 - Cardinality - M:N
 - Degree = 2 (Binary)
 - Participation Constraint - meta reality(1, M); photos(1, N)
- Supervises [SELF-RELATIONSHIP]
 - Every employee has a supervisor, who itself an employee.
 - One employee can be supervised by only one supervisor, but an employee/supervisor can supervise many different employees.
 - Cardinality - 1:N
 - Degree = 2 (Binary)
 - Participation Constraint - Employee[Supervisor] (0,N); Employee[Supervisee] (1,1)

Functional Requirements -

- MODIFICATIONS:
 - Insert-
 - Add entry of a new user into the database of **IIITVR**, when a user signs up.
 - Add a new entry in the chats schema when a message is sent.
 - Add a new entry in the photos table when a user posts a new photo.
 - Make a new issue table entity when a user opens an issue.
 - Delete-
 - Remove the details of a plan after the offer period is done.
 - Remove the list of issues that have been resolved from the issues schema.
 - Remove the user details if they delete the account.
 - Removal of employee details that he left the company.
 - Remove the photo if the user wants to delete it permanently.
 - Update-
 - Change the user's plan when the user switches the plan or when it expires.
 - Update the preferences of a certain user based on his current expectations.
 - Update plans if any employee updates any plan.
- RETRIEVALS:
 - Aggregates-
 - Total cost earned by the company in a given period - sum
 - the Average number of messages a user sends per day - average
 - Oldest and youngest user of the service - max and min
 - Projection-
 - The number of issues having the same tag on a given day.
 - The number of users using the most basic and highest plans.
 - List of users having no messages in the last three months (Customer Retention)
 - Selection-
 - User having more than a specific number of issues in a certain period
 - Employees resolving the most number of issues for a particular tag in a week.
 - Users having a plan above a certain value (for premium services)
 - Search-
 - List of users whose plan is going to expire in 7 days
 - The list of photos that were uploaded over two years ago.
 - List of users having at least one issue not resolved.
 - Analysis-
 - The average duration of enrollment of the users for each plan available. This can be used to modify the rates accordingly.
 - Customer Acquisition rate - New Customers joined after a particular date
 - Customer Retention rate - Based on the number of Customers still present on the platform after a particular date
 - the Average number of chats per user per month - To see how active the users are generally.

Summary-

- This database has detailed and structured information for the **IIITVR** mini-world.
- This database will enable fast access to a profile for the users and easy maintenance for the maintainers.
- This data will also help them understand the volume of data flow and prepare their servers accordingly.
- This database will immensely help the founders of **IIITVR** to grow their platform to the biggest.

Reference-> https://youtu.be/9R013apT_Pw

📺 Something Like Love, a short film starring Shantanu Maheshwari & Shei...