Community Connect – Investigation/Database Design

Investigation

Summaries of two articles discussing the challenges of volunteer management and the importance of community engagement

1. Challenges of Volunteer Management

Article: "The 5 Biggest Volunteer Management Challenges You Might Face"

- **Retention:** Keeping volunteers engaged and committed over time is a significant struggle for many organizations. Lack of recognition, insufficient communication, or unclear roles can lead to high turnover.
- **Undesirable Behaviour:** Volunteer managers often face challenges enforcing rules when volunteers display disruptive behaviour. The risk is that enforcing policies might cause volunteers to withdraw, yet ignoring behaviour can harm team morale. Key solutions involve fostering perceived organizational support and fulfilling psychological contracts—ensuring volunteers feel valued and understood.
- **Resource Constraints:** Lack of administrative support, training, and recognition programs can hinder efforts to recruit and retain volunteers effectively.
- **Sustainability:** Volunteer burnout, lack of professional management, and insufficient support systems can impact the sustainability of volunteer programs. Effective management practices are essential to address these issues.

2. Importance of Community Engagement

Article: "Why Community Engagement Matters" (Penn State University)

- **Trust and Communication:** Engaging communities increases trust in organizations and governance. Inclusive partnerships can address issues more effectively and foster mutual understanding.
- **Effective Solutions:** Involving community members in projects leads to practical, widely accepted outcomes since diverse local knowledge is utilized.
- **Empowerment and Skills:** Community engagement empowers individuals, integrates people from varied backgrounds, and helps citizens develop problem-solving and communication abilities.

- **Networks and Prevention:** It builds local social networks and offers multiple opportunities for people to discuss concerns, allowing early identification and resolution of issues before they escalate.
- **Sustained Success:** Ongoing engagement supports the development of local leadership, strengthens accountability, and enhances the sustainability of community-led projects.

1. https://www.rosterfy.com/blog/the-5-biggest-volunteer-management-challenges-you-might-face
2. https://aese.psu.edu/research/centers/cecd/engagement-toolbox/engagement/why-community-engagement-matters

Project Plan/Timeline

Week 1: Investigation & Design (Aug 18 – Aug 24, 2025)

- Research volunteer platforms and summarise findings.
- Identify entities, attributes, and relationships.
- Develop ERD (Crow's Foot notation, minimum 5 tables, including an M:N relationship).
- Perform normalisation to 3NF and produce relational notation.
- Create a full data dictionary.
- Begin research into ethical and legal considerations (Australian Privacy Principles).

Deliverables: Project management plan, research summary, ERD, normalisation documentation, data dictionary.
Milestone: Database design finalised.

Week 2: Database Implementation & Basic Application (Aug 25 – Aug 31, 2025)

- Build SQLite database with CREATE TABLE and constraints.
- Populate with 15–20 realistic records.
- Set up Flask web app and connect database.
- Implement CRUD functionality (Create volunteer, Read organisations, Update volunteer contact, Delete event).
- Write journal entry on ACID properties.

Deliverables: SQL script, SQLite database file, Flask app files, CRUD screenshots, journal entry.
Milestone: Working prototype with CRUD operations.

Week 3: Advanced Queries & Final Report (Sept 1 – Sept 6, 2025)

- Implement advanced queries in Flask (search by skill, volunteer sign-ups per event, statistics with aggregates, formatted volunteer list with age).
- Write a short discussion on data quality and cleaning.
- Compile final report including design docs, SQL, screenshots, evaluation, and ethical/legal/security issues (APP5, APP10, APP11, APP12).

Deliverables: Final Flask app, SQLite database, comprehensive report.
Milestone: Fully functional system with advanced features and documentation submitted.

6th September 2025 – final application and report due.

Deconstruction

Core Entities	Attributes
Volunteers	**VolunteerID** (Primary Key) - FirstName - LastName - Email - Password - Phone - Address - DateOfBirth - Availability (True/False) - ProfilePhoto - EmergencyContact
Organisations	**OrganisationID** (Primary Key) - Name - Description - ContactPerson - Email - Password - Phone - Address - Website (optional) - Logo (optional)
Events	**EventID** (Primary Key) - Name - Description - Date - StartTime - EndTime - Location - Status (e.g., Upcoming, Ongoing, Completed, Cancelled)
Roles	**RoleID** (Primary Key) - Name - Description - Status (e.g., pending, accepted, denied)
Skills	**SkillID** (Primary Key) - Name - Description (optional)
Entity Relationships	Volunteer -> Skill (M:N) Volunteer -> Event (M:N) Event -> Organisation (M:1) Event -> Skills (M:N) Volunteer -> Roles (M:N) Events -> Roles (M:N)

Normalisation

Note:
Underline indicates Primary Key (PK) (e.g. SkillID)
Italicised + Underline indicates Foreign Key (*FK*) (e.g. *SkillID* FK Skills)

0NF Starting Point

(**VolunteerID**, FirstName, LastName, Password, Email, Phone, Address, DateOfBirth, Availability, ProfilePhoto, EmergencyContact, **SkillID**, SkillName, SkillDescription, **OrganisationID**, OrganisationName, OrganisationDescription, OrganisationContactPerson, OrganisationEmail, OrganisationPassword, OrganisationPhone, OrganisationAddress, OrganisationWebsite, OrganisationLogo, **EventID**, EventName, EventDescription, EventDate, StartTime, EndTime, EventLocation, EventStatus, Roles, RolesDescription)

1NF Normalisation

Table does not satisfy requirements for 1NF normalisation as 'Roles' stores non-atomic values. This can be resolved to 1NF by ensuring all attributes store atomic values. (i.e. 'Roles' -> Role using individual rows for each skill attribute)

Resolved 1NF Relational Notation:

(**VolunteerID**, FirstName, LastName, Password, Email, Phone, Address, DateOfBirth, Availability, ProfilePhoto, EmergencyContact, **SkillID**, SkillName, SkillDescription, OrganisationID, OrganisationName, OrganisationDescription, OrganisationContactPerson, OrganisationEmail, OrganisationPassword, OrganisationPhone, OrganisationAddress, OrganisationWebsite, OrganisationLogo, **EventID**, EventName, EventDescription, EventDate, StartTime, EndTime, EventLocation, EventStatus, **RoleID**, Role, RoleDescription, RoleStatus)

2NF Normalisation

Table does not satisfy requirements for 2NF normalisation due to the presence of partial dependencies; there is a quadruple composite key (VolunteerID, SkillID, EventID, RoleID) (OrganisationID not required as EventID -> OrganisationID is M:1)

VolunteerID alone identifies a volunteer, but a volunteer can have many skills and participate in many events associated with different organisations.
SkillID alone identifies a skill, but many volunteers can have many different skills across many events.
EventID identifies the event, but many events can have many skills and volunteers.

RoleID identifies the role of a volunteer, but many volunteers can have many roles in many events.

(OrganisationID depends on **EventID** and thus all attributes depending on OrganisationID have transitive dependencies.)

To resolve this, all attributes depending only on 1 part of the composite key must be separated into their own respective tables. This requires creation of multiple junction tables resolving M:N relationships to ensure no partial dependencies.

M:N Data Relationships:

Volunteers -> Skills (M:N)
Events -> Skills (M:N)
Events -> Volunteers (M:N)

And thus **partial dependencies** are as follows:

FirstName, LastName, Email, Password, Phone, Address, DateOfBirth, Availability, ProfilePhoto, EmergencyContact -> **VolunteerID**

EventName, EventDescription, EventDate, StartTime, EndTime, EventLocation, EventStatus, OrganisationID -> **EventID**

SkillName, SkillDescription -> **SkillID**

RoleName, RoleDescription -> **RoleID**

Resolved 2NF Relational Notation:

Volunteers (**VolunteerID**, FirstName, LastName, Email, Password, Phone, Address, DateOfBirth, Availability, ProfilePhoto, EmergencyContact)

VolunteerSkills(***VolunteerID*** **FK Volunteers**, ***SkillID*** **FK Skills**)

Skills (**SkillID**, Name, Description)

Organisations (**OrganisationID**, Name, Email, Password, Description, ContactPerson, Email, Phone, Address, Website, Logo)

EventSkills(***EventID*** **FK Events**, ***SkillID*** **FK Skills**)

Signups(**ID**, ***EventID*** **FK Events**, ***VolunteerID*** **FK Volunteers**, ***RoleID*** **FK Roles**, Status)

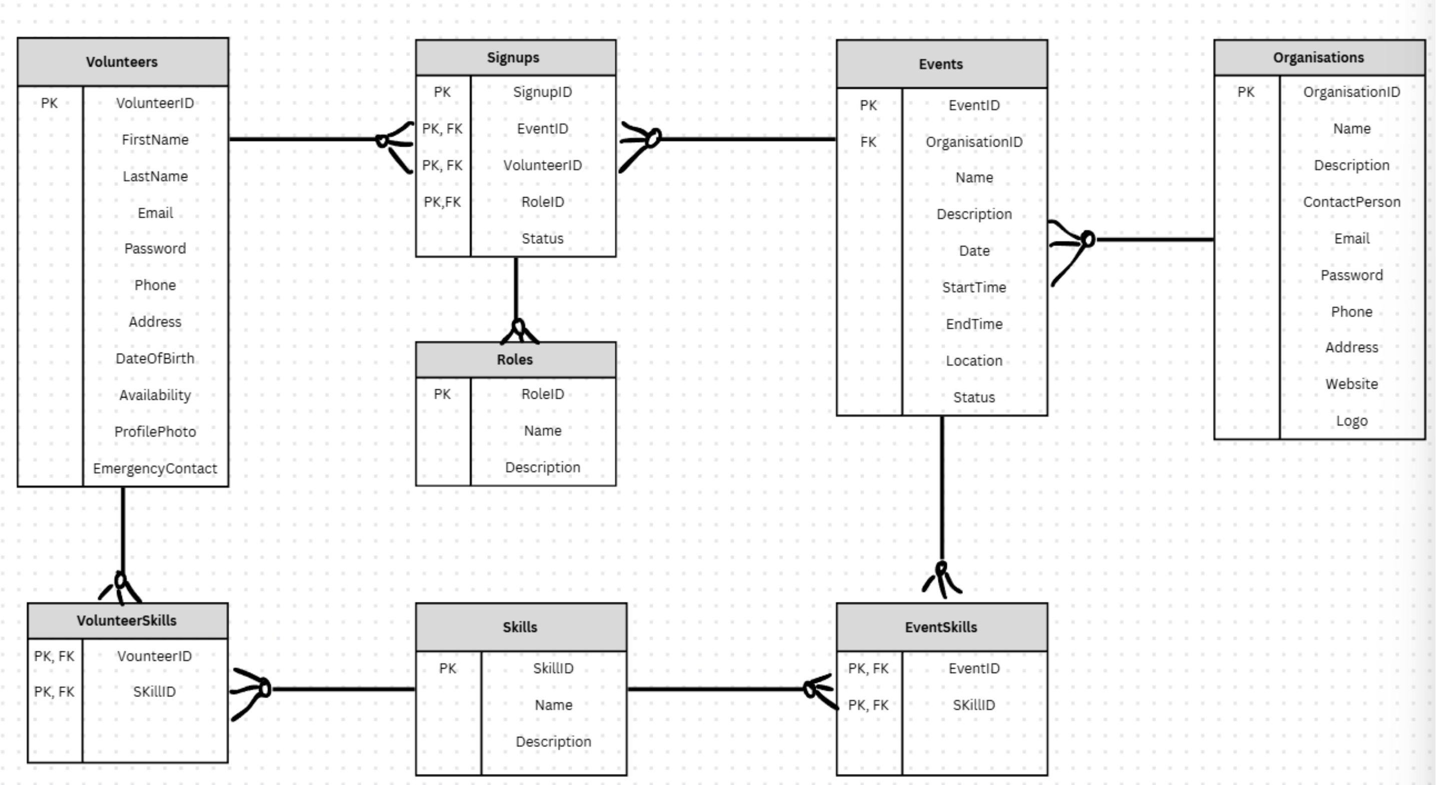
Roles(**RoleID**, Name, Description)

Events (**EventID**, ***OrganisationID*** **FK Organisations**, Name, Description, Date, StartTime, EndTime, Location, Status)

3NF Normalisation

For a database to satisfy 3NF normalisation, it must already be in 2NF, as well as have no transitive dependencies. All transitive dependencies have been eliminated, and thus the schema is already in 3NF. (Organisation attributes -> OrganisationID -> **EventID** transitive dependency has been resolved)

ERD



Volunteers	
PK	VolunteerID
	FirstName
	LastName
	Email
	Password
	Phone
	Address
	DateOfBirth
	Availability
	ProfilePhoto
	EmergencyContact

Signups	
PK	SignupID
PK, FK	EventID
PK, FK	VolunteerID
PK,FK	RoleID
	Status

Events	
PK	EventID
FK	OrganisationID
	Name
	Description
	Date
	StartTime
	EndTime
	Location
	Status

Organisations	
PK	OrganisationID
	Name
	Description
	ContactPerson
	Email
	Password
	Phone
	Address
	Website
	Logo

Roles	
PK	RoleID
	Name
	Description

VolunteerSkills	
PK, FK	VounteerID
PK, FK	SKillID

Skills	
PK	SkillID
	Name
	Description

EventSkills	
PK, FK	EventID
PK, FK	SKillID

Data Dictionaries

Volunteers

ElementName	DataType	Size	Description	Constraints
VolunteerID	Integer		Unique Identifier for Volunteer	PK, Unique, Not Null
FirstName	Text	20	First Name of Volunteer	Not Null
LastName	Text	20	Last Name of Volunteer	Not Null
Email	Text	320	Email Address	Unique, Not Null
Password	Text	128	Hashed Password	Not Null
Phone	Text	32	Phone Number	
Address	Text	255	Home Address	
DateOfBirth	Date		Date of Birth	
Availability	Boolean		Availability Status	
ProfilePhoto	BLOB		Profile Photo	
EmergencyContact	Text	32	Emergency Contact Number	

Events

ElementName	DataType	Size	Description	Constraints
EventID	Integer		Unique identifier for Event	PK, Unique, Not Null
OrganisationID	Integer		Unique identifier for the Organising Organisation	FK → Organisations
Name	Text	100	Name of Event	Not Null
Description	Text	255	Description of Event	
Date	Date		Date of Event	
StartTime	Time		Start Time	
EndTime	Time		End Time	
Location	Text	255	Location of Event	
Status	Text	20	Status (Upcoming, etc.)	

Organisations

ElementName	DataType	Size	Description	Constraints
OrganisationID	Integer		Unique Identifier for Organisation	PK, Unique, Not Null
Name	Text	100	Organisation Name	Not Null
Description	Text	255	Organisation Description	
ContactPerson	Text	50	Contact Person	
Email	Text	320	Contact Email	Unique, Not Null
Password	Text	128	Hashed Password	Not Null
Phone	Text	32	Contact Phone	
Address	Text	255	Address	
Website	Text	255	Website URL	
Logo	BLOB		Organisation Logo	

Skills

ElementName	DataType	Size	Description	Constraints
SkillID	Integer		Unique identifier for Skill	PK, Unique, Not Null
Name	Text	50	Name of Skill	Not Null
Description	Text	255	Description of Skill	

Roles

ElementName	DataType	Size	Description	Constraints
RoleID	Integer		Unique identifier for Role	PK, Unique, Not Null
Name	Text	50	Name of Role	Not Null
Description	Text	255	Description of Role	

Signups

ElementName	DataType	Size	Description	Constraints
SignupID	Integer		Unique identifier for Signup	PK, Unique, Not Null
EventID	Integer		Unique identifier for Event being signed up for	FK → Events, Not Null
VolunteerID	Integer		Volunteer signing up	FK → Volunteers, Not Null
RoleID	Integer		Role assigned	FK → Roles, Nullable
Status	Text	20	Status (Pending, Confirmed, etc.)	

EventSkills

ElementName	DataType	Size	Description	Constraints
EventID	Integer		Unique identifier for Event needing the skill	PK, FK → Events, Not Null
SkillID	Integer		Unique identifier for Required skill for the event	PK, FK → Skills, Not Null

(Composite PK: EventID + SkillID)

VolunteerSkills

ElementName	DataType	Size	Description	Constraints
VolunteerID	Integer		Unique identifier for Volunteer with this skill	PK, FK → Volunteers, Not Null
SkillID	Integer		Unique identifier for Skill of the volunteer	PK, FK → Skills, Not Null

(Composite PK: VolunteerID + SkillID)