

Segment 1

Case description: Volunteer Information System

Willow Creek Academy is a public charter school that offers a core knowledge curriculum to grades K to 8 with two classes per grade consisting of 25 students per class. As a charter school, it lacks the resources provided to traditional public schools and requires parental involvement through various volunteer activities. The school senate is interested in a computerized system to reduce the effort to track volunteer activities and communicate volunteer information to parents. To streamline operations, the school wants to augment the current manual system by providing the ability to create and remove ad hoc volunteer areas throughout the school year, maintain calendars for repetitive volunteer areas, send reminders to parents with few or no volunteer hours, audit volunteer hours, and integrate information about families that is outside of the information currently stored in the manual volunteer information system. The current manual system revolves around the volunteer book, a binder with one page per family, as its central repository. The lunchroom and traffic areas have signup documents and calendars for scheduling. The volunteer coordinators maintain most of the information in the manual system through informal communication and non-standard documents. Willow Creek Academy needs a computerized system to manage volunteers more efficiently.

Workflow

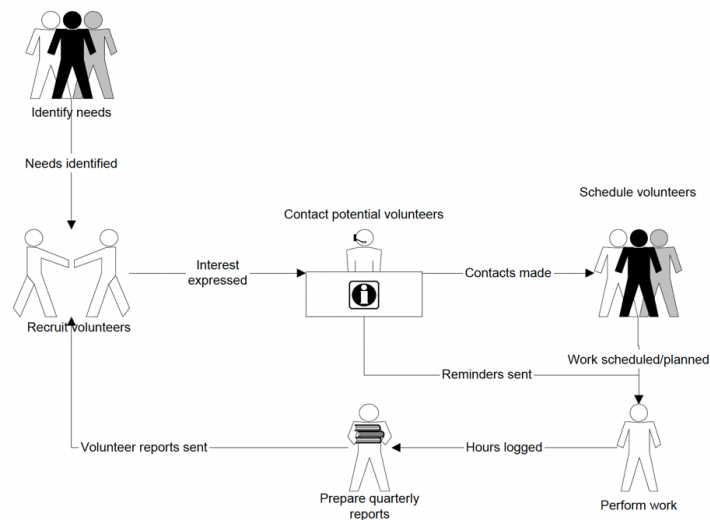


Figure 1: Typical Workflow for Volunteer Information and Processing

Business Rules

1. A volunteer may volunteer for one or more events during the school year.
2. A volunteer is associated with only one family.
3. A volunteer area is controlled by one Family.
4. A volunteer area may have one or more families.
5. A family may have one or more volunteers working for them.

Conceptual data models (i.e., ERDs, etc.)

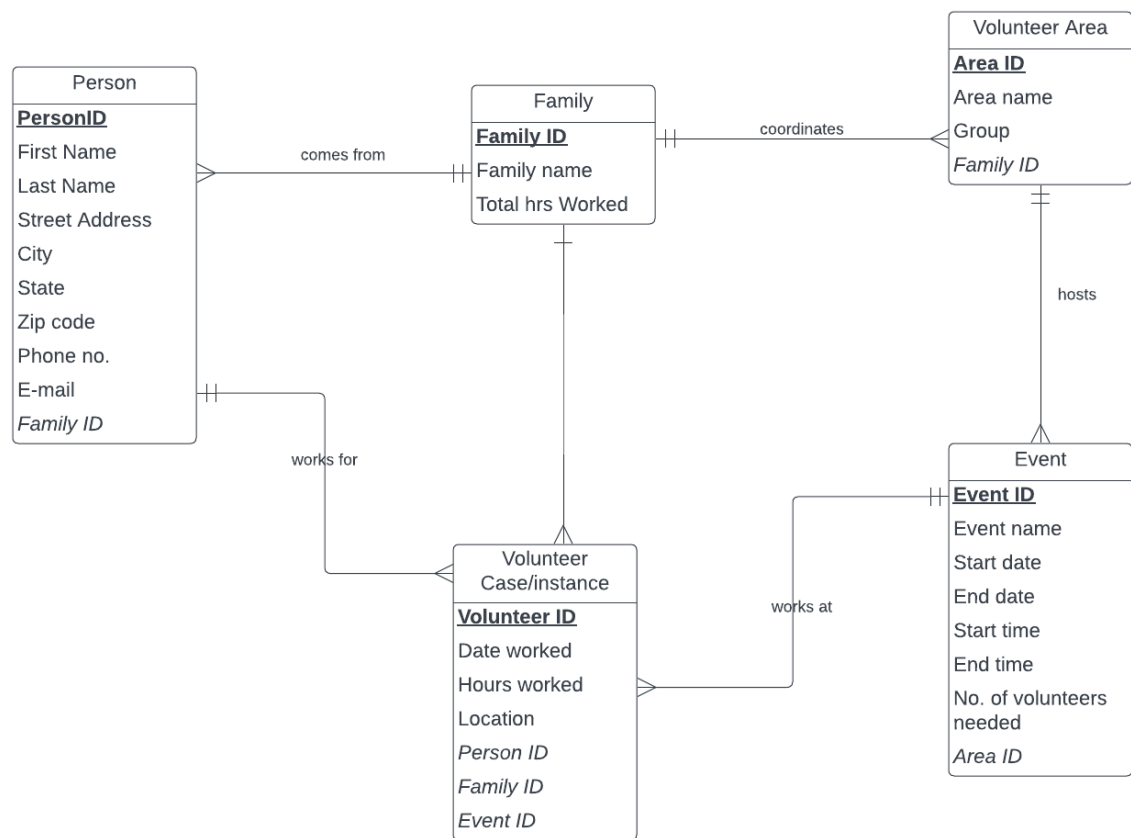


Figure 2: ERD for the Volunteer Information System

Schema

family(family_id, family_name, total_hours_worked)
Area(area_id, area_name, control_group, *family_id*)
Event (event_id, event_name, start_date, end_date, volunteers_needed, *area_id*)
Person(person_id, first_name, last_name, street_address, city, state, zip_code, phone_number, email, *family_id*)
VolunteerCase(Volunteer_id, date_worked, hours_worked, location, *person_id*, *family_id*, *event_id*)

Populated data in tables

The screenshot displays the Oracle SQL Developer environment. On the left, the 'Connections' pane shows a connection to 'Volunteer'. The 'Tables (Filtered)' list includes AREA, EVENT, FAMILY, PERSON, and VOLUNTEERCASE. The main window shows a SQL script in the 'Worksheet' tab. The script includes several SQL statements: a SELECT from event, an INSERT INTO Person, a SELECT from Person, an INSERT INTO VolunteerCase, a SELECT from VolunteerCase, and a series of DELETE statements for Event, Area, Person, Family, and VolunteerCase. Below the script, the 'Query Result' pane shows the results of a SELECT query. The results are displayed in a table with columns AREA_ID, AREA_NAME, CONTROL_GROUP, and FAMILY_ID. The data shows five rows of information about different areas.

AREA_ID	AREA_NAME	CONTROL_GROUP	FAMILY_ID
1	100 Classroom help	Parent teacher association	8
2	101 Lunchroom	faculty	15
3	102 Traffic	Parent teacher association	1
4	103 Front desk	faculty	20
5	104 Work day	senate	5

Figure 3: Area table

The screenshot shows the SQL Developer interface with the 'Event' table selected in the 'Tables (Filtered)' pane. The SQL editor contains a query to select all data from the 'Event' table. The bottom pane displays the query results for the 'Event' table.

EVENT_ID	EVENT_NAME	START_DATE	END_DATE	VOLUNTEERS_NEEDED	AREA_ID
1	581WCA_1	03-JAN-22 09.00.00.000000000 AM	05-JAN-22 02.30.00.000000000 PM	15	100
2	582WCA_2	05-FEB-22 07.00.00.000000000 AM	15-FEB-22 09.30.00.000000000 AM	12	102
3	583WCA_3	10-JAN-22 11.00.00.000000000 AM	25-JAN-22 12.30.00.000000000 PM	18	101
4	584WCA_4	15-MAR-22 06.30.00.000000000 AM	16-MAR-22 02.30.00.000000000 PM	30	104
5	585WCA_5	01-MAR-22 09.00.00.000000000 AM	31-MAR-22 04.30.00.000000000 PM	25	104
6	586WCA_6	01-APR-22 03.00.00.000000000 PM	15-APR-22 04.30.00.000000000 PM	13	102
7	587WCA_7	16-AUG-22 07.30.00.000000000 AM	30-AUG-22 02.30.00.000000000 PM	22	103
8	588WCA_8	09-FEB-22 11.00.00.000000000 AM	20-FEB-22 02.45.00.000000000 PM	45	104
9	589WCA_9	02-SEP-22 01.15.00.000000000 PM	17-SEP-22 03.15.00.000000000 PM	5	100
10	510WCA_10	01-MAY-22 11.00.00.000000000 AM	01-JUN-22 12.30.00.000000000 PM	14	101

Figure 4: Event table

The screenshot shows the SQL Developer interface with the 'Family' table selected in the 'Tables (Filtered)' pane. The SQL editor contains a query to select all data from the 'Family' table. The bottom pane displays the query results for the 'Family' table.

FAMILY_ID	FAMILY_NAME	TOTAL_HOURS_WORKED
1	1 Stout	(null)
2	2 Branch	(null)
3	3 Yates	(null)
4	4 Elliott	(null)
5	5 Haynes	(null)
6	6 Morrow	(null)
7	7 Bailey	(null)
8	8 Hayes	(null)
9	9 Shepard	(null)
10	10 Boyle	(null)
11	11 Fowler	(null)
12	12 Castillo	(null)
13	13 Holt	(null)

Figure 5 : Family table

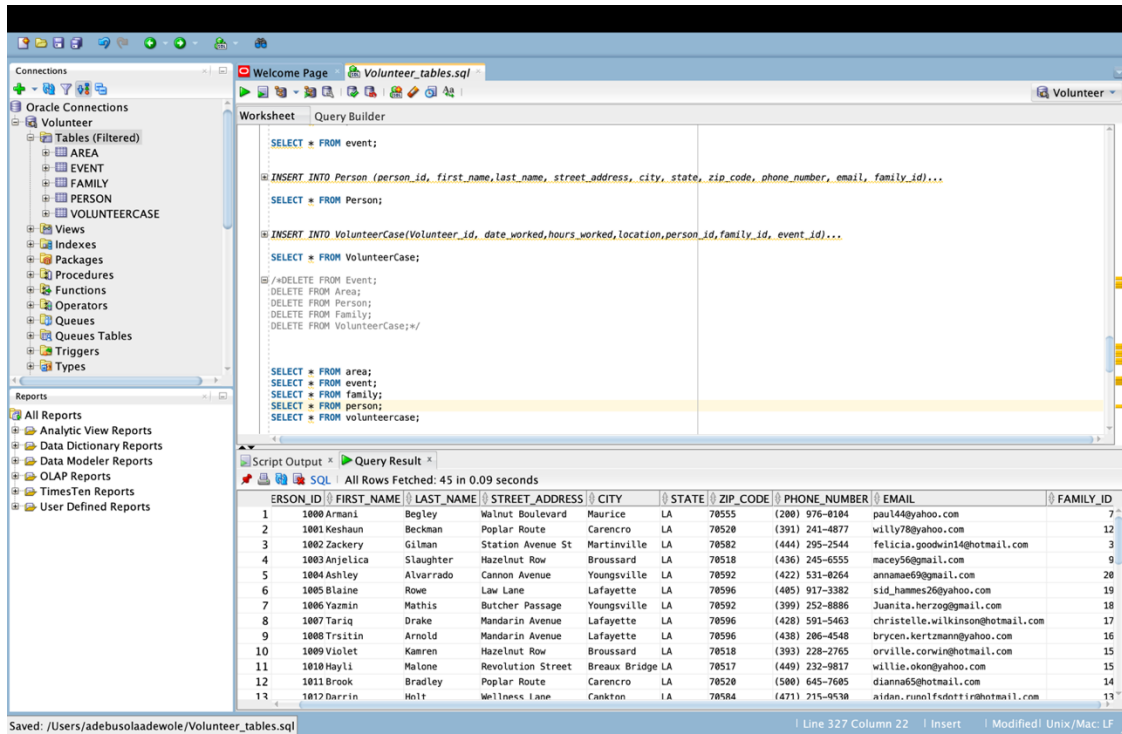


Figure 6: Person table

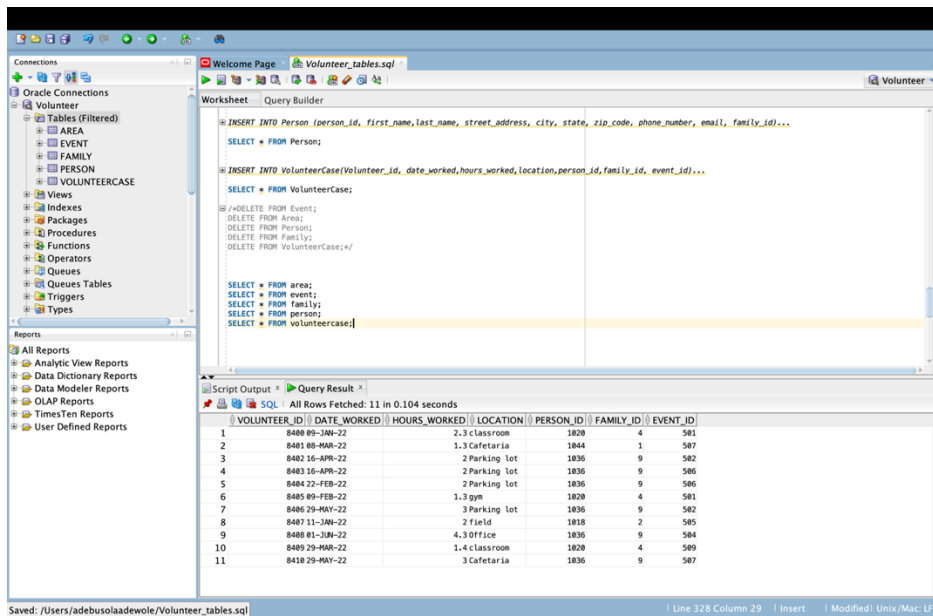


Figure 7: VolunteerCase table

Technical Manual (report) for DB backend

Family & Volunteer report:

Select statement to find all the volunteers that have worked under family 020

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with 'Volunteer' selected. The main pane shows a SQL query in the 'Query Builder' tab:

```
SELECT DISTINCT a.volunteer_id, a.family_id, b.first_name, b.last_name, b.person_id
FROM volunteercase a
JOIN Person b ON a.person_id = b.person_id
WHERE a.family_id= 020;
```

The 'Script Output' pane at the bottom shows the query results:

VOLUNTEER_ID	FAMILY_ID	FIRST_NAME	LAST_NAME	PERSON_ID
1	8481	20 Winter	Stout	1844
2	8487	20 Juanita	Branch	1818
3	8410	20 Iris	Shepard	1836

JOIN the family table to find out who family 020 is

The screenshot shows the Oracle SQL Developer interface with a more complex query in the 'Query Builder' tab:

```
SELECT DISTINCT a.volunteer_id, a.family_id, b.first_name, b.last_name, b.person_id, c.family_name
FROM volunteercase a
JOIN Person b ON a.person_id = b.person_id
JOIN family c ON c.family_id= a.family_id
WHERE c.family_id= 020;
```

The 'Script Output' pane at the bottom shows the query results:

VOLUNTEER_ID	FAMILY_ID	FIRST_NAME	LAST_NAME	PERSON_ID	FAMILY_NAME
1	8481	20 Winter	Stout	1844	Alvarado
2	8487	20 Juanita	Branch	1818	Alvarado
3	8410	20 Iris	Shepard	1836	Alvarado

From the screenshot Family Alvarado of those three volunteers during the event

Find out how many hours Family 014 has worked for the 2022 school year?

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with 'Volunteer' selected. The main window is in 'Query Builder' mode. The SQL statement in the worksheet is:

```
SELECT family_id, SUM(hours_worked) as total_hours_worked
FROM volunteercase
GROUP BY family_id
HAVING family_id = 014;
```

The 'Query Result' pane at the bottom shows the output of the query:

FAMILY_ID	TOTAL_HOURS_WORKED
14	6.6

The status bar at the bottom indicates the file is saved at `/Users/adebusolaadewole/Volunteer_tables.sql` and the cursor is at Line 359, Column 24.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with 'Volunteer' selected. The main window is in 'Query Builder' mode. The SQL statement in the worksheet is:

```
SELECT a.family_id, SUM(a.hours_worked) as total_hours_worked, b.family_name
FROM volunteercase a
JOIN family b ON a.family_id = b.family_id
GROUP BY a.family_id, b.family_name
HAVING a.family_id = 014;
```

The 'Query Result' pane at the bottom shows the output of the query:

FAMILY_ID	TOTAL_HOURS_WORKED	FAMILY_NAME
14	6.6	Bradley

The status bar at the bottom indicates the file is saved at `/Users/adebusolaadewole/Volunteer_tables.sql` and the cursor is at Line 360, Column 26.

The Bradley family has worked for 6.6 hours so far.

Event report:

Find volunteers who worked during event 506?

The screenshot shows a SQL IDE interface with the following components:

- Connections:** A tree view on the left showing a connection to a database named 'Volunteer'.
- Tables (Filtered):** A list of tables including AREA, EVENT, FAMILY, PERSON, and VOLUNTEERCASE.
- Views:** A list of views including Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, Types, and Sequences.
- Script Output:** A tab showing the SQL query being executed.
- Query Result:** A tab showing the results of the query, which are 5 rows of volunteer data.

The SQL query in the Script Output tab is:

```
SELECT a.volunteer_id, a.event_id, a.date_worked, b.first_name, b.last_name
FROM volunteercase a
JOIN person b ON a.family_id = b.family_id
WHERE a.event_id = 506;
```

The Query Result tab displays the following data:

	VOLUNTEER_ID	EVENT_ID	DATE_WORKED	FIRST_NAME	LAST_NAME
1	8404	506	22-FEB-22	Zackery	Gilman
2	8403	506	16-APR-22	Jerod	Fowler
3	8404	506	22-FEB-22	Graciela	Yates
4	8403	506	16-APR-22	Aliya	Fowler
5	8404	506	22-FEB-22	Luisa	Yates

The status bar at the bottom indicates the file is saved at `/Users/adebusolaadewole/Volunteer_tables.sql` and shows the current cursor position at Line 354, Column 1.