Help Desk Schedule system

By - Natnael Mengistu

Table of Contents

Executive Summary --------- 3

ER Diagram -------- 4

Student Employee Table -------- 5

Availability Table ------- 6

Schedule Table -------- 7

Supervisor Table --------- 8

Location Table -------- 9

Slot Table -------- 10

Logic Table ------- 11

Views ------- 12

Triggers -------- 13

Security -------- 14

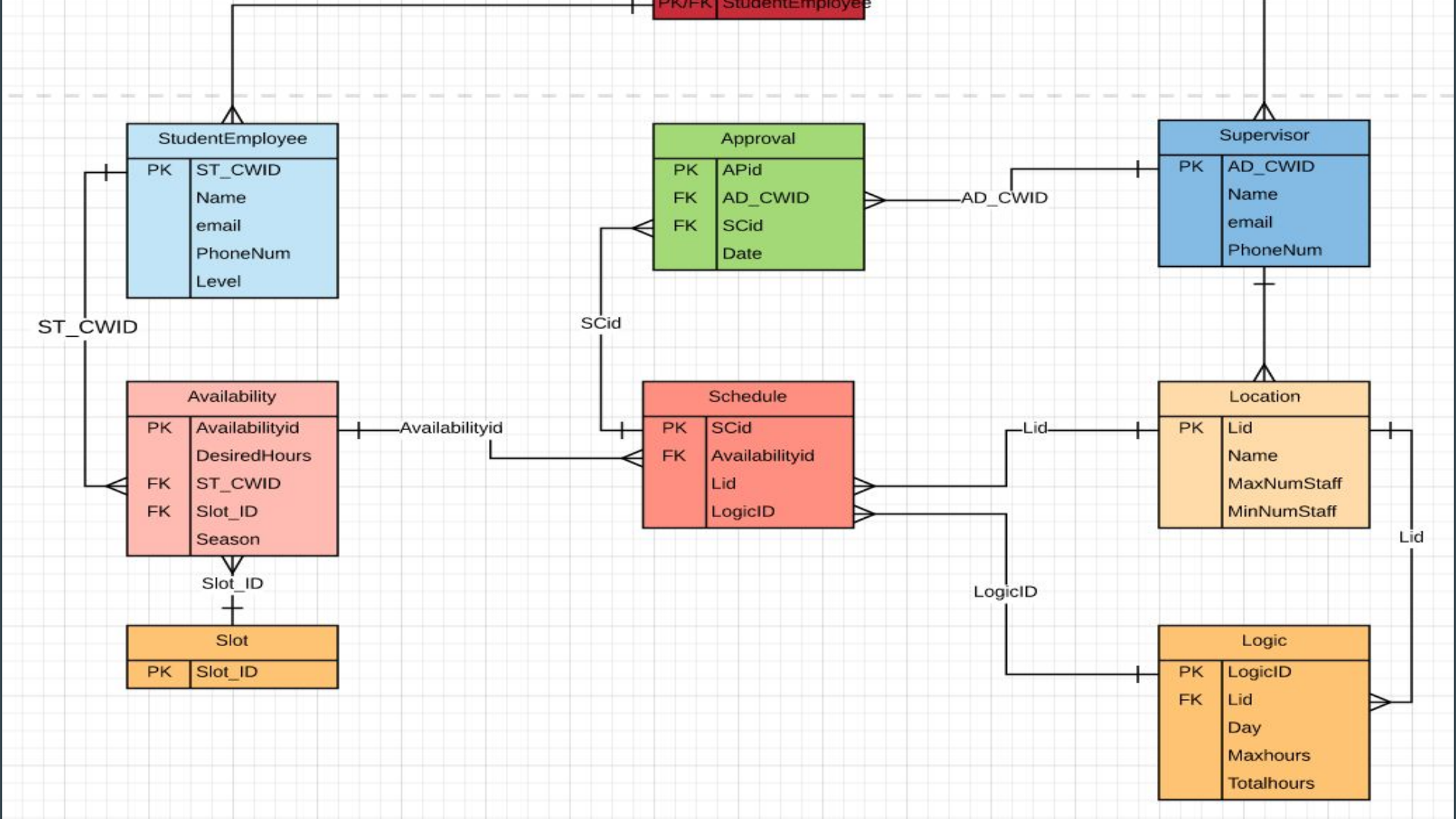
Implementation -------- 15

Executive Summary:

Project Goal:

- To develop a database for a scheduling system
- Quick and easy access to employee's Information
- Easy to implement new conditions

This document provides a new database system to address the tedious process of schedule making. This database system makes updating and inserting new data simpler and avoids Duplicate or separate data lists.


PK/FK StudentEmployee
StudentEmployee
PK ST_CWID
Name
email
PhoneNum
Level
Approval
PK APid
FK AD_CWID
FK SCid
Date
AD_CWID
Supervisor
PK AD_CWID
Name
email
PhoneNum
ST_CWID
SCid
Availability
PK Availabilityid
DesiredHours
FK ST_CWID
FK Slot_ID
Season
Availabilityid
Schedule
PK SCid
FK Availabilityid
Lid
LogicID
Lid
Location
PK Lid
Name
MaxNumStaff
MinNumStaff
Lid
Slot_ID
Slot
PK Slot_ID
LogicID
Logic
PK LogicID
FK Lid
Day
Maxhours
Totalhours

Student Employee Table

This table is where basic information about a student employee is stored . Every student has unique id(ST_CWID)

```
CREATE TABLE studentEmployee (

  ST_CWID  char(10)  PRIMARY KEY ,

  Name   text,

  Email   char(50),

  PhoneNum int,

  Level  text

);
```

Functional Dependencies

ST_CWID ⟶ Name , email,PhoneNum , Level

Data Output | Explain | Messages | History

st_cwid character	name text	email character	phonenum integer	level text
S79878909	MIKE	mike123...	845545656	Trainee
S11122233	Frank	frankyboy...	656237242	Experienc...
S51434365	Joey	jman@ho...	2134564568	Experienc...
S09337890	Mary	themary4...	845213202	Trainee
S79668919	Nora	nora.lee1...	848787877	Experienc...

Availability Table

The supervisor will share the link or form with the student employees, the student employees will access the link to open up Availability form. Once Student workers accesses the form, they fill the slots and submit it. Every submitted form has its own id (Availibilityid).

```
CREATE TABLE  Availability(

Availabilityid char(10) PRIMARY KEY,

DesiredHours  int,

ST_CWID char(10) not null references StudentEmployee(ST_CWID),

Slot_ID char(10) not null references Slot(Slot_ID),

Season text

        );
```

Functional Dependencies

Availabilityid ⟶ DesiredHours,ST_CWID,Slot_ID,Season

Data Output | Explain | Messages | History

availabilit... character	desiredh... integer	st_cwid character	slot_id character	season text
A001	6	S79878909	SL333	Fall
A002	3	S11122233	SL334	Spring
A003	5	S51434365	SL335	Fall
A004	2	S09337890	SL336	Spring
A005	1	S79878909	SL338	Fall

Schedule Table

Collect all the availability forms for each one of the student worker and generate one single schedule satisfying the conditions and logics.

```
CREATE TABLE  Schedule(

SCid char(10) PRIMARY KEY,

Availabilityid char(10),

Lid char(10),

LogicID char(10)

);
```

Functional Dependencies

SCid ⟶ Availabilityid , Lid ,LogicID

Data Output | Explain | Messages | History

scid character	availabilit... character	lid character	logicid character
S877937...	AV4867738	ST3093	AV0250
S453625...	AV1645238	DY3092	AV0249
S209394...	AV7849874	DN558	AV0249
S465123...	AV0293589	LT4948	AV0249

Supervisor Table

This table is where basic information about the Supervisor is stored . Every supervisor has unique id(AD_CWID)

Supervisor table

```
CREATE TABLE  Supervisor(
  AD_CWID char(10)  PRIMARY KEY,
  Name  text,
  Email char(50),
  PhoneNum int
);
```

Functional Dependencies

AD_CWID ⟶ Name,Email,PhoneNum

Data Output | Explain | Messages | History

ad_cwid character	name text	email character	phonenum integer
AD001	Alan	Labouseu...	516652556
AD002	Kathy	kathy1@...	845789465

Location Table

```
CREATE TABLE  Location(

Lid text PRIMARY KEY,

Name text,

MaxNumStaff int,

MinNumStaff  int

);
```

Functional Dependencies

Lid ⟶ Name,MaxNumStaff,MinNumStaff

Data Output | Explain | Messages | History

lid text	name text	maxnum... integer	minnums... integer
DN558	Lobby	2	1
LT9898	Raised flo...	3	2
LT4948	Lounge	2	1
ST3093	gym	4	2
SC87	Library	2	1
DY3092	Cafe	1	0

Slot Table

Every slot represents one hour and every Slot has a primary key, This makes data retrieval simpler

```
CREATE TABLE  Slot(

Slot_ID char(10) PRIMARY KEY,

);
```

Functional Dependencies

Slot_ID ⟶

Data Output | Explain | Messages | History

slot_id character
SL333
SL334
SL335
SL336

Logic Table

Logics Table holds specific conditions depending on the location and day. On this create statement we also have CHECK constraints which satisfy some of the conditions specified by the Helpdesk.

```
CREATE TABLE  Logic(

LogicID char(10) PRIMARY KEY,

Lid  char(10) not null references Location(Lid),

Day text,

Maxhours int CHECK (Maxhours <= 8),

Totalhours int CHECK (Totalhours <= 40)

);
```

Functional Dependencies

Lid ⟶ Logic,LogicID,Day,Maxhours,Totalhours

Data Output | Explain | Messages | History

logicid character	lid character	day text	maxhours integer	totalhours integer
L0001	SC87	Monday	8	40
L0003	ST3093	Wednesday	8	40
L0005	DN558	Saturday	4	8

Views

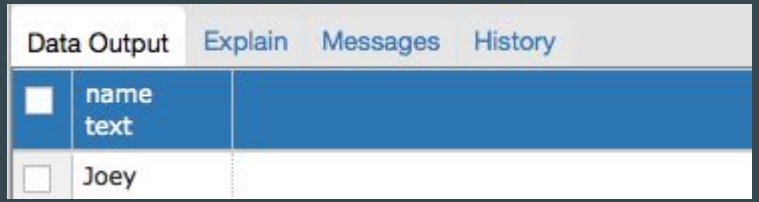
Supervisor can generate the following report to track which student worker is ready to work for specific desired hours depending on the requirements.

EXAMPLE- Let's assume the supervisor is looking for someone to cover 5 hr shift

```
CREATE VIEW View_Name AS

SELECT Name

FROM StudentEmployee, Availability

WHERE StudentEmployee.ST_CWID = Availability.ST_CWID

AND Desiredhours > 5 ;
```

Query

```
SELECT * FROM View_Name
```



Trigger

Checks if the Desired hour is under 40 before updating the Table

```
CREATE TRIGGER MaxhourTrigger

BEFORE UPDATE DesiredHours ON Availability

REFERENCING

  OLD ROW AS OLDTuple

  NEW ROW AS NEWTuple

FOR EACH ROW

WHEN ( 40 < (select DesiredHours from Availability))

INSERT INTO Availability

END;
```

SECURITY

Supervisors Alan and Kathy have full access to the Schedule

```
CREATE ROLE Admin1

GRANT SELECT ,INSERT,UPDATE

ON Schedule TO Alan;

        CREATE ROLE Admin2

        GRANT SELECT, INSERT, UPDATE

        ON Schedule to Kathy;
```

Implementation Notes

Purpose :

The purpose of this Database is to make a schedule based on the Availability of employees and conditions set by the Department. Also keeps the school's student worker law by keeping work hours in check.

Test Data :

Test Data user in the Database is randomly generated to satisfy the Referential Integrity.

Known Issue

- Trigger - we have to update the table . But CHECK constraint is keeping the Tables integrity.
- The conditions might change from time to time so Logic table needs updating

Future Enhancement

- Front and Backend page for data entry and Access.
- More data Sample / Helpdesk employee’s information will be added in the system