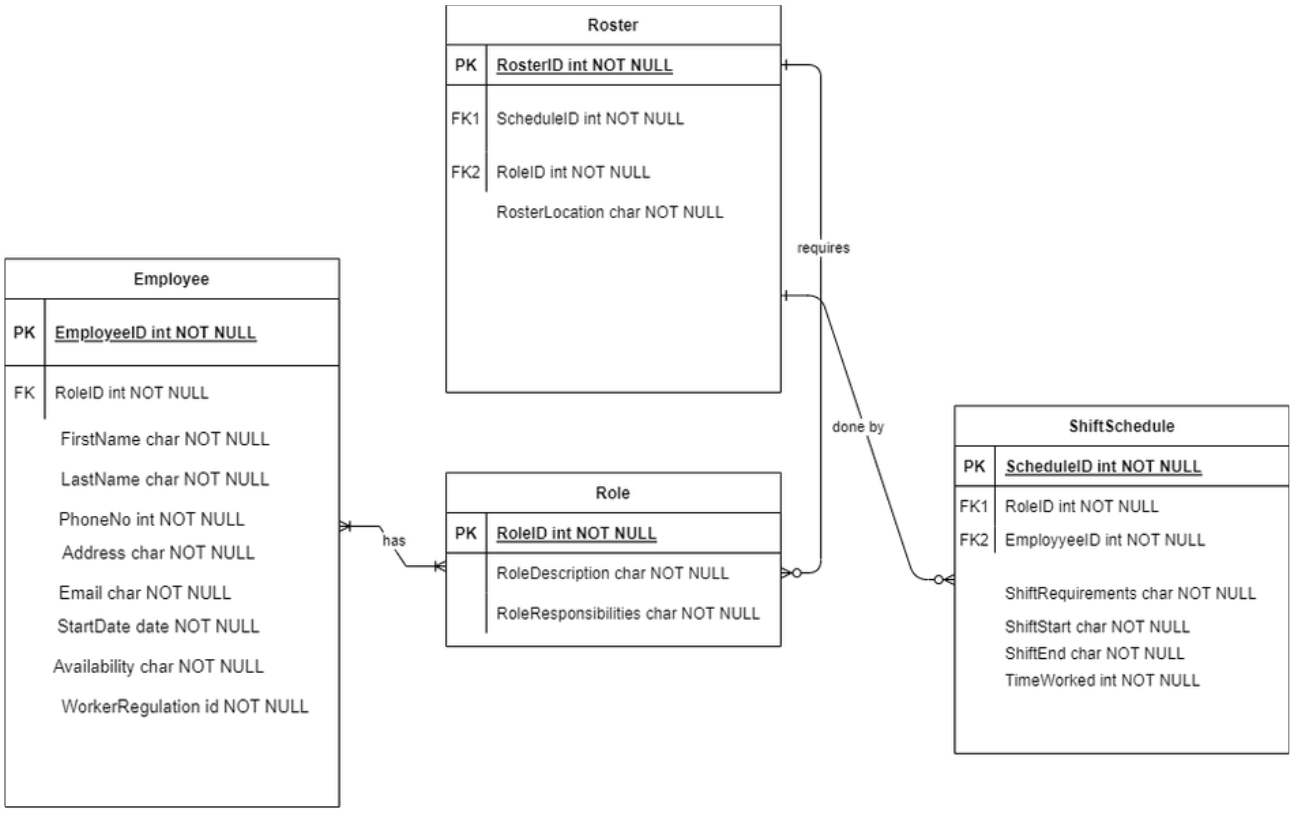
Initial ER Diagram

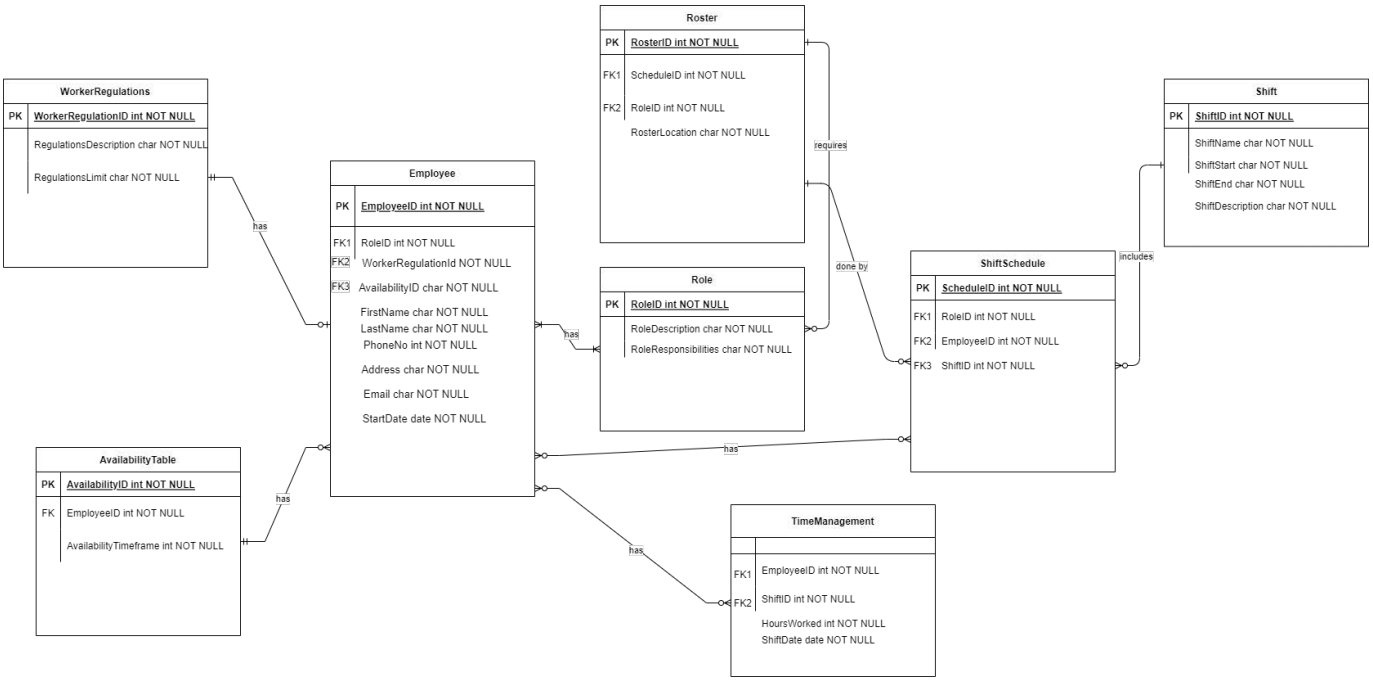
Initial ERD made in week 4

The image below is the ER diagram made initially without normalisation as a blueprint for the database.



ERD Diagram

After careful review and multiple iterations and normalisation of the ER diagram, this is the final ERD for the database created.



Physical database

availabilityTable
employee
role
roster
shift
shiftSchedule
timeManagement
workerRegulation

phpMyAdmin was the database creation used for this project. Each of the tables in the ER Diagram have been curated in this with sample dummy data inserted to check for data authenticity and for SQL use cases to be tested.

Entire database file:

s103834192_db(2).sql
27 May 2023, 04:38 AM

Availability Table

availabilityID	employeeID	availabilityTimeframe

SQL statements

Create Table

```
CREATE TABLE availabilityTable (
  availabilityID int(5) NOT NULL,
  employeeID int(5) NOT NULL,
  availabilityTimeframe varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Indexes and keys

```
ALTER TABLE availabilityTable
  ADD PRIMARY KEY (availabilityID),
  ADD KEY employeeID (employeeID);
```

Constraints

```
ALTER TABLE availabilityTable
  ADD CONSTRAINT fk_availabilityTable_employee FOREIGN KEY (employeeID) REFERENCES employee (employeeID);
```

Auto-Increment

```
ALTER TABLE availabilityTable
  MODIFY availabilityID int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=70
```

Employee Table

employeeID	roleID	firstName	lastName	phoneNumber	email	address	startDate	availabilityID	workerRegulationID

SQL statements

Create Table

```
CREATE TABLE employee (
  employeeID int(5) NOT NULL,
  roleID int(5) NOT NULL,
  firstName varchar(20) NOT NULL,
  lastName varchar(20) NOT NULL,
  phoneNumber varchar(10) NOT NULL,
  email varchar(25) NOT NULL,
  address varchar(25) NOT NULL,
  startDate varchar(10) NOT NULL,
  availabilityID int(5) NOT NULL,
  workerRegulationID int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE employee
  ADD PRIMARY KEY (employeeID),
  ADD KEY roleID (roleID),
  ADD KEY workerRegulationID (workerRegulationID),
  ADD KEY availabilityID (availabilityID);
```

Constraints

```
ALTER TABLE employee
  ADD CONSTRAINT fk_employee_availability FOREIGN KEY (availabilityID) REFERENCES availabilityTable
(availabilityID),
  ADD CONSTRAINT fk_employee_role FOREIGN KEY (roleID) REFERENCES role (roleID),
  ADD CONSTRAINT fk_employee_workerRegulation FOREIGN KEY (workerRegulationID) REFERENCES workerRegulation
(workerRegulationID);
```

Auto-Increment

```
ALTER TABLE employee
  MODIFY employeeID int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

Role Table

roleID	roleDescription	roleResponsibility

SQL statements

Create Table

```
CREATE TABLE role (
  roleID int(4) NOT NULL DEFAULT '0',
  roleDescription varchar(50) NOT NULL,
  roleResponsibility varchar(149) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE role
  ADD PRIMARY KEY (roleID);
```

Roster Table

SQL statements

Create Table

```
CREATE TABLE roster (
  rosterID int(5) NOT NULL,
  scheduleID int(5) NOT NULL,
  roleID int(5) NOT NULL,
  rosterLocation varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE roster
  ADD PRIMARY KEY (rosterID),
  ADD KEY scheduleID (scheduleID),
  ADD KEY roleID (roleID);
```

Constraints

```
ALTER TABLE roster
  ADD CONSTRAINT fk_roster_role FOREIGN KEY (roleID) REFERENCES role (roleID),
  ADD CONSTRAINT fk_roster_schedule FOREIGN KEY (scheduleID) REFERENCES shiftSchedule (scheduleID);
```

Auto-Increment

```
ALTER TABLE roster
  MODIFY rosterID int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

Shift Table

SQL statements

Create Table

```
CREATE TABLE shift (
  shiftID int(4) NOT NULL,
  shiftName varchar(50) NOT NULL,
  shiftStart datetime NOT NULL,
  shiftEnd datetime NOT NULL,
  shiftDescription varchar(149) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE shift
  ADD PRIMARY KEY (shiftID);
```

Auto-Increment

```
ALTER TABLE shift
  MODIFY shiftID int(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

Shift Schedule Table

SQL statements

Create Table

```
CREATE TABLE shiftSchedule (
  scheduleID int(5) NOT NULL,
  roleID int(5) NOT NULL,
  employeeID int(5) NOT NULL,
  shiftID int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE shiftSchedule
  ADD PRIMARY KEY (scheduleID),
  ADD KEY roleID (roleID),
  ADD KEY employeeID (employeeID),
  ADD KEY shiftID (shiftID);
```

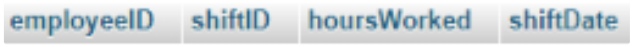
Constraints

```
ALTER TABLE shiftSchedule
  ADD CONSTRAINT fk_shiftSchedule_shift FOREIGN KEY (shiftID) REFERENCES shift (shiftID),
  ADD CONSTRAINT fk_shiftSchedule_employee FOREIGN KEY (employeeID) REFERENCES employee (employeeID),
  ADD CONSTRAINT fk_shiftSchedule_role FOREIGN KEY (roleID) REFERENCES role (roleID);
```

Auto-Increment

```
ALTER TABLE shiftSchedule
  MODIFY scheduleID int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

Time Management Table



SQL statements

Create Table

```
CREATE TABLE timeManagement (
  employeeID int(5) NOT NULL,
  shiftID int(5) NOT NULL,
  hoursWorked int(5) NOT NULL,
  shiftDate date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE timeManagement
  ADD KEY employeeID (employeeID),
  ADD KEY shiftID (shiftID);
```

Constraints

```
ALTER TABLE timeManagement
  ADD CONSTRAINT fk_timeManagement_shift FOREIGN KEY (shiftID) REFERENCES shift (shiftID),
  ADD CONSTRAINT fk_timeManagement_employee FOREIGN KEY (employeeID) REFERENCES employee (employeeID);
```

Worker Regulation Table

SQL statements

Create Table

```
CREATE TABLE workerRegulation (
  workerRegulationID int(5) NOT NULL,
  regDescription varchar(50) NOT NULL,
  regLimit varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Indexes and keys

```
ALTER TABLE workerRegulation
  ADD PRIMARY KEY (workerRegulationID);
```

Auto-Increment

```
ALTER TABLE workerRegulation
  MODIFY workerRegulationID int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

Data creation and Null values

Cobbl.io was used to create dummy data for the database. Each table used data that followed the constraints in regards to each attribute's **data type** and followed **uniqueness** for each cell record. Afterwards, the following statements were used:

Availability Table

availabilityID	employeeID	availabilityTimeframe
1	1	6am - 3pm
2	2	1pm - 9pm
3	3	10am - 4pm
4	4	11am - 5pm
5	5	9am - 2pm
6	6	12pm - 6pm
7	7	4pm - 9pm
8	8	8am - 4pm
9	9	8.30am - 5pm
10	10	7am - 2pm

Dummy Data Insertion

Insert Data

```
INSERT INTO availabilityTable (availabilityID, employeeID, availabilityTimeframe) VALUES
(1, 1, '6am - 3pm'),
(2, 2, '1pm - 9pm'),
(3, 3, '10am - 4pm'),
(4, 4, '11am - 5pm'),
(5, 5, '9am - 2pm'),
(6, 6, '12pm - 6pm'),
(7, 7, '4pm - 9pm'),
(8, 8, '8am - 4pm'),
(9, 9, '8.30am - 5pm'),
(10, 10, '7am - 2pm');
```

Employee Table

employeeID	roleID	firstName	lastName	phoneNumber	email	address	startDate	availabilityID	workerRegulationID
1	1	Jack	Simmons	636-090-66	jsimmons@aol.com	126 Brookside Drive	28/2/2011	1	1
2	2	Sofia	Young	802-526-53	smyoung@aol.com	86733 Meadow Lane	6/6/2013	2	2
3	3	Gabriel	Parker	557-375-45	gabriel.joseph.parker@aol	643 Myrtle Avenue	9/2/2012	3	3
4	4	Christopher	Foster	251-033-74	christopherfoster@ymail.c	89 3rd Street West	8/4/2021	4	4
5	5	Eric	Stewart	254-950-23	eric.f.stewart@rocketmail	8268 South Street	2/10/2021	5	5
6	6	Kelsey	Jackson	251-169-16	kelsey.f.jackson@yahoo.co	67 King Street	10/7/2015	6	5
7	7	Victoria	Richardson	971-644-98	victoriarichardson@aol.co	571 6th Street West	16/12/2014	7	3
8	8	Charles	Martin	423-835-36	charles.martin@outlook.co	4601 Water Street	17/9/2015	8	2
9	9	Robert	Coleman	617-501-47	robert.joseph.coleman@yma	71 Dogwood Drive	26/12/2010	9	1
10	10	Julian	Smith	813-002-31	julian_g_smith10@rocketma	2457 Railroad Street	16/3/2020	10	3
11	7	Tumbleweed	Tumbleweed	918-937-11	j_a_walker@hotmail.com	84 Division Street	1/05/2014	9	2
12	6	Independence	Independence	360-733-27	alexandramwilson@rocketma	9192 Main Street North	4/09/2021	6	2
13	8	Unbleached silk	Unbleached silk	872-062-19	egray54@aol.com	82430 Bridge Street	19/01/2018	2	5
14	8	Silver chalice	Silver chalice	220-052-51	e_russell@gmail.com	129 Dogwood Lane	17/04/2014	9	4
15	2	Purpureus	Purpureus	915-154-27	a_w_sanders@yahoo.com	9528 Riverside Drive	11/02/2014	3	2
16	4	Ochre	Ochre	701-349-39	j_bryant22@live.com	1067 Cambridge Court	14/12/2014	6	3
17	2	Mauve taupe	Mauve taupe	906-915-42	kayla_garcia@ymail.com	22 Route 29	7/01/2014	4	5
18	6	Hollywood cerise	Hollywood cerise	816-210-77	j.d.diaz@ymail.com	393 Chestnut Street	8/02/2015	2	2
19	4	Green	Green	405-443-04	r.w@hotmail.com	16 Willow Street	20/04/2018	1	3
20	9	Vermilion	Vermilion	386-750-72	aubrey.james@gmail.com	39 Dogwood Drive	9/06/2014	8	3
21	1	Unmellow yellow	Unmellow yellow	814-878-60	saraanncoleman@outlook.co	93142 Dogwood Lane	9/12/2015	8	1
22	1	Rich black	Rich black	203-014-23	danielcarter@aol.com	44 Spruce Street	13/08/2021	2	4
23	7	Banana Mania	Banana Mania	919-874-31	claire_k_thompson@outlook	402 Park Avenue	7/04/2012	7	2
24	3	Yellow	Yellow	313-741-60	c.f@ymail.com	3943 Route 10	30/09/2017	3	1
25	1	Old lace	Old lace	405-792-65	aturner39@hotmail.com	35190 Oak Lane	30/01/2021	5	3

Dummy Data Insertion

Insert Data

```
INSERT INTO employee (employeeID, roleID, firstName, lastName, phoneNumber, email, address, startDate,
availabilityID, workerRegulationID) VALUES
(1, 1, 'Jack', 'Simmons', '636-090-66', 'jsimmons@aol.com', '126 Brookside Drive', '28/2/2011', 1, 1),
(2, 2, 'Sofia', 'Young', '802-526-53', 'smyoung@aol.com', '86733 Meadow Lane', '6/6/2013', 2, 2),
(3, 3, 'Gabriel', 'Parker', '557-375-45', 'gabriel.joseph.parker@aol', '643 Myrtle Avenue', '9/2/2012', 3, 3),
(4, 4, 'Christopher', 'Foster', '251-033-74', 'christopherfoster@ymail.c', '89 3rd Street West', '8/4/2021', 4, 4),
(5, 5, 'Eric', 'Stewart', '254-950-23', 'eric.f.stewart@rocketmail', '8268 South Street', '2/10/2021', 5, 5),
(6, 6, 'Kelsey', 'Jackson', '251-169-16', 'kelsey.f.jackson@yahoo.co', '67 King Street', '10/7/2015', 6, 6),
(7, 7, 'Victoria', 'Richardson', '971-644-98', 'victoriarichardson@aol.co', '571 6th Street West', '16/12/2014', 7,
7),
(8, 8, 'Charles', 'Martin', '423-835-36', 'charles.martin@outlook.co', '4601 Water Street', '17/9/2015', 8, 8),
```

```
(9, 9, 'Robert', 'Coleman', '617-501-47', 'robert.joseph.coleman@yma', '71 Dogwood Drive', '26/12/2010', 9, 9),
(10, 10, 'Julian', 'Smith', '813-002-31', 'julian_g_smith10@rocketma', '2457 Railroad Street', '16/3/2020', 10, 10);
```

Role Table

roleID	roleDescription	roleResponsibility
1	Web Developer	Develop and maintain web applications and websites...
2	Sales Representative	Sell products or services to potential customers b...
3	Project Manager	Plan, execute and finalize projects according to s...
4	Human Resources Coordinator	Assist with recruitment, employee relations, and H...
5	Marketing Manager	Develop and implement marketing strategies to prom...
6	Data Scientist	Analyze large datasets using statistical and machi...
7	Financial Analyst	Provide financial analysis and reporting to suppor...
8	Customer Service Representative	Respond to customer inquiries and issues via phone...
9	Operations Manager	Oversee day-to-day operations of a business or dep...
10	Graphic Designer	Create visual designs for marketing materials, web...

Dummy Data Insertion

Insert Data

```
INSERT INTO role (roleID, roleDescription, roleResponsibility) VALUES
(1, 'Web Developer', 'Develop and maintain web applications and websites using various programming languages and tools'),
(2, 'Sales Representative', 'Sell products or services to potential customers by making presentations, following up on leads, and closing deals'),
(3, 'Project Manager', 'Plan, execute and finalize projects according to strict deadlines and within budget, while ensuring all stakeholders are kept informed'),
(4, 'Human Resources Coordinator', 'Assist with recruitment, employee relations, and HR administration, including maintaining employee records and benefits information'),
(5, 'Marketing Manager', 'Develop and implement marketing strategies to promote products or services and increase brand awareness'),
(6, 'Data Scientist', 'Analyze large datasets using statistical and machine learning techniques to identify patterns and insights that drive business decisions'),
(7, 'Financial Analyst', 'Provide financial analysis and reporting to support budgeting, forecasting, and decision-making processes'),
(8, 'Customer Service Representative', 'Respond to customer inquiries and issues via phone, email, or chat, and provide timely and effective resolution'),
(9, 'Operations Manager', 'Oversee day-to-day operations of a business or department, ensuring efficient and effective use of resources to achieve organizational goals'),
(10, 'Graphic Designer', 'Create visual designs for marketing materials, websites, and other digital or print media');
```

Roster Table

rosterID	scheduleID	roleID	rosterLocation
1	1	1	Showell
2	2	2	Montour Falls
3	3	3	Old Lyme
4	4	4	Raleigh
5	5	5	Rio Grande City
6	6	6	Bowling Green
7	7	7	Cayey
8	8	8	Boston
9	9	9	Rockport
10	10	10	Milwaukee

Dummy Data Insertion

Insert Data

```
INSERT INTO roster (rosterID, scheduleID, roleID, rosterLocation) VALUES
(1, 1, 1, 'Showell'),
(2, 2, 2, 'Montour Falls'),
(3, 3, 3, 'Old Lyme'),
(4, 4, 4, 'Raleigh'),
(5, 5, 5, 'Rio Grande City'),
(6, 6, 6, 'Bowling Green'),
(7, 7, 7, 'Cayey'),
(8, 8, 8, 'Boston'),
(9, 9, 9, 'Rockport'),
(10, 10, 10, 'Milwaukee');
```

Shift Table

shiftID	shiftName	shiftStart	shiftEnd	shiftDescription
1	Help Desk Shift	2023-04-28 07:00:00	2023-04-28 18:13:24	Respond to user inquiries and resolve technical is...
2	Network Operations Center (NOC) Shift	2023-04-28 07:00:00	2023-04-28 19:13:24	Monitor network and system performance, respond to...
3	Security Operations Center (SOC) Shift	2023-04-28 07:00:00	2023-04-28 20:13:24	Monitor security events and alerts, investigate po...
4	Application Support Shift	2023-04-28 06:30:00	2023-04-28 23:30:39	Provide technical support for enterprise applicati...
5	Database Administrator Shift	2023-04-28 06:15:00	2023-04-28 18:00:00	Manage and maintain databases, perform backup and ...
6	Infrastructure Shift	2023-04-28 08:00:00	2023-04-28 20:15:00	Install, configure, and maintain hardware and soft...
7	DevOps Shift	2023-04-28 07:40:00	2023-04-28 15:30:00	Build, test, and deploy software releases, manage ...
8	QA Analyst Shift	2023-04-28 09:30:00	2023-04-28 17:00:00	Perform testing and quality assurance activities, ...
9	Cloud Operations Shift	2023-04-28 10:00:00	2023-04-28 18:00:00	Manage and maintain cloud infrastructure, monitor ...
10	Project Management Shift	2023-04-28 07:00:24	2023-04-28 18:07:24	Plan and coordinate IT projects, manage project ti...

Dummy Data Insertion

Insert Data

```
INSERT INTO shift (shiftID, shiftName, shiftStart, shiftEnd, shiftDescription) VALUES
(1, 'Help Desk Shift', '2023-04-28 07:00:00', '2023-04-28 18:13:24', 'Respond to user inquiries and resolve technical issues via phone, email, or chat, and escalate complex issues to the appropriate teams'),
(2, 'Network Operations Center (NOC) Shift', '2023-04-28 07:00:00', '2023-04-28 19:13:24', 'Monitor network and system performance, respond to alerts and alarms, and troubleshoot and resolve issues as needed'),
(3, 'Security Operations Center (SOC) Shift', '2023-04-28 07:00:00', '2023-04-28 20:13:24', 'Monitor security events and alerts, investigate potential security incidents, and respond to security breaches'),
(4, 'Application Support Shift', '2023-04-28 06:30:00', '2023-04-28 23:30:39', 'Provide technical support for enterprise applications, troubleshoot issues, and collaborate with developers and other teams to resolve issues'),
(5, 'Database Administrator Shift', '2023-04-28 06:15:00', '2023-04-28 18:00:00', 'Manage and maintain databases, perform backup and recovery tasks, and optimize database performance'),
(6, 'Infrastructure Shift', '2023-04-28 08:00:00', '2023-04-28 20:15:00', 'Install, configure, and maintain hardware and software components, and respond to system failures or outages'),
(7, 'DevOps Shift', '2023-04-28 07:40:00', '2023-04-28 15:30:00', 'Build, test, and deploy software releases, manage infrastructure as code, and automate development processes'),
(8, 'QA Analyst Shift', '2023-04-28 09:30:00', '2023-04-28 17:00:00', 'Perform testing and quality assurance activities, including writing and executing test cases, identifying defects, and verifying fixes'),
(9, 'Cloud Operations Shift', '2023-04-28 10:00:00', '2023-04-28 18:00:00', 'Manage and maintain cloud infrastructure, monitor resource utilization, and optimize costs'),
(10, 'Project Management Shift', '2023-04-28 07:00:24', '2023-04-28 18:07:24', 'Plan and coordinate IT projects, manage project timelines and budgets, and communicate project status to stakeholders');
```

Shift Schedule Table

scheduleID	roleID	employeeID	shiftID
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10

Dummy Data Insertion

Insert Data

```
INSERT INTO shiftSchedule (scheduleID, roleID, employeeID, shiftID) VALUES
(1, 1, 1, 1),
(2, 2, 2, 2),
(3, 3, 3, 3),
(4, 4, 4, 4),
(5, 5, 5, 5),
(6, 6, 6, 6),
(7, 7, 7, 7),
(8, 8, 8, 8),
(9, 9, 9, 9),
(10, 10, 10, 10);
```

Time Management Table

employeeID	shiftID	hoursWorked	shiftDate
1	1	21	2023-04-02
2	2	24	2023-03-03
3	3	19	2023-02-15
4	4	25	2023-04-01
5	5	15	2023-01-21
6	6	32	2023-04-13
7	7	41	2023-03-04
8	8	27	2023-03-23
9	9	18	2023-02-08
10	10	33	2023-04-04

Dummy Data Insertion

Insert Data

```
INSERT INTO timeManagement (employeeID, shiftID, hoursWorked, shiftDate) VALUES
(1, 1, 21, '2023-04-02'),
(2, 2, 24, '2023-03-03'),
(3, 3, 19, '2023-02-15'),
(4, 4, 25, '2023-04-01'),
(5, 5, 15, '2023-01-21'),
(6, 6, 32, '2023-04-13'),
(7, 7, 41, '2023-03-04'),
(8, 8, 27, '2023-03-23'),
(9, 9, 18, '2023-02-08'),
(10, 10, 33, '2023-04-04');
```

Worker Regulation Table

workerRegulationID	regDescription	regLimit
1	Full Time	40 hours per week
2	Part Time	20 hours per week
3	Casual	30 hours per week
4	Trainee	8 hours per week
5	Overtime	40+ hours per week

Dummy Data Insertion

Insert Data

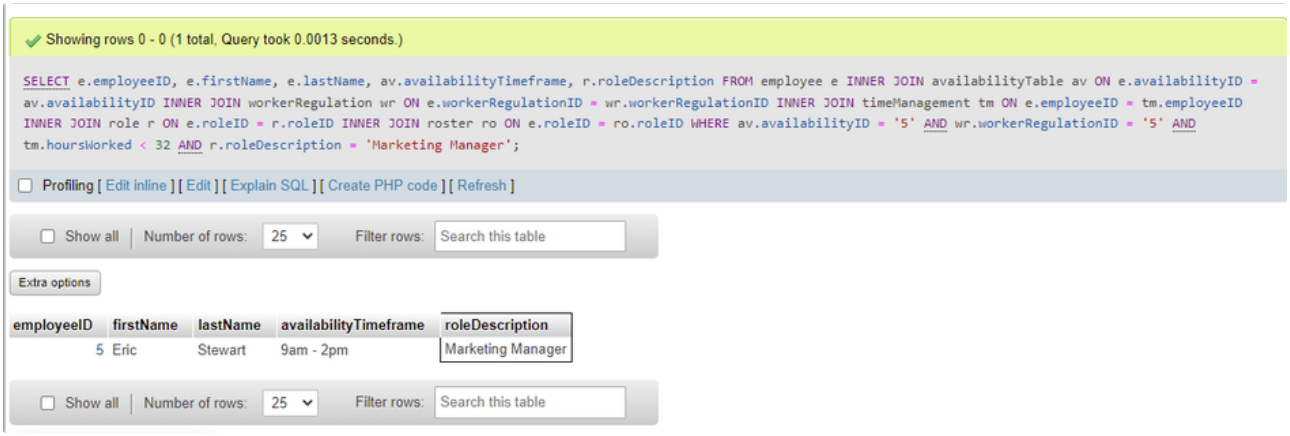
```
INSERT INTO workerRegulation (workerRegulationID, regDescription, regLimit) VALUES
(1, 'Full Time', '40 hours per week'),
(2, 'Part Time', '20 hours per week'),
(3, 'Casual', '30 hours per week'),
(4, 'Trainee', '8 hours per week'),
(5, 'Overtime', '40+ hours per week');
```

Use cases and SQL statements

Using SQL queries to find employees to be rostered into a certain shift

To find employees can work in the morning, have worked less than 32 hours, are marketing managers and can include overtime workers:

```sql
SELECT e.employeeID, e.firstName, e.lastName, av.availabilityTimeframe, r.roleDescription
FROM employee e
INNER JOIN availabilityTable av ON e.availabilityID = av.availabilityID
INNER JOIN workerRegulation wr ON e.workerRegulationID = wr.workerRegulationID
INNER JOIN timeManagement tm ON e.employeeID = tm.employeeID
INNER JOIN role r ON e.roleID = r.roleID
INNER JOIN roster ro ON e.roleID = ro.roleID
WHERE av.availabilityID = '5'
  AND wr.workerRegulationID = '5'
  AND tm.hoursWorked < 32
  AND r.roleDescription = 'Marketing Manager';
```



To find employees have worked less than 45 hours, can work in the afternoon, only casual workers, can be financial analysts or project managers or operations managers:

```sql
SELECT e.employeeID, e.firstName, e.lastName, av.availabilityTimeframe, s.shiftName, r.roleDescription,
tm.hoursWorked, wr.regDescription
FROM employee e
INNER JOIN availabilityTable av ON e.availabilityID = av.availabilityID
INNER JOIN workerRegulation wr ON e.workerRegulationID = wr.workerRegulationID
INNER JOIN timeManagement tm ON e.employeeID = tm.employeeID
INNER JOIN role r ON e.roleID = r.roleID
INNER JOIN shiftSchedule ss ON e.employeeID = ss.employeeID
INNER JOIN shift s ON ss.shiftID = s.shiftID
WHERE tm.hoursWorked < 45
  AND av.availabilityID = '7'
  AND wr.workerRegulationID = '3'
  AND r.roleDescription IN ('Financial Analyst', 'Project Manager', 'Operations Manager');
```

Showing rows 0 - 0 (1 total, Query took 0.0015 seconds.)

```
SELECT e.employeeID, e.firstName, e.lastName, av.availabilityTimeframe, s.shiftName, r.roleDescription, tm.hoursWorked, wr.regDescription FROM employee e INNER
JOIN availabilityTable av ON e.availabilityID = av.availabilityID INNER JOIN workerRegulation wr ON e.workerRegulationID = wr.workerRegulationID INNER JOIN
timeManagement tm ON e.employeeID = tm.employeeID INNER JOIN role r ON e.roleID = r.roleID INNER JOIN shiftSchedule ss ON e.employeeID = ss.employeeID INNER JOIN
shift s ON ss.shiftID = s.shiftID WHERE tm.hoursWorked < 45 AND av.availabilityID = '7' AND wr.workerRegulationID = '3' AND r.roleDescription IN ('Financial
Analyst', 'Project Manager', 'Operations Manager');
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

employeeID	firstName	lastName	availabilityTimeframe	shiftName	roleDescription	hoursWorked	regDescription
7	Victoria	Richardson	4pm - 9pm	DevOps Shift	Financial Analyst	41	Casual

To find employees have worked for more than 5 years, can be web developers or financial analysts or customer service representatives and only part-time workers:

```sql
SELECT e.employeeID, e.firstName, e.lastName, e.startDate,r.roleDescription, wr.regDescription
FROM employee e
INNER JOIN role r ON e.roleID = r.roleID
INNER JOIN workerRegulation wr ON e.workerRegulationID = wr.workerRegulationID
WHERE TIMESTAMPDIFF(YEAR, STR_TO_DATE(e.startDate, '%d/%m/%Y'), CURDATE()) > 5
  AND r.roleDescription IN ('Web Developer', 'Financial Analyst', 'Customer Service
Representative')
  AND wr.workerRegulationID = '2';

```

Showing rows 0 - 4 (5 total, Query took 0.0016 seconds.)

```
SELECT e.employeeID, e.firstName, e.lastName, e.startDate,r.roleDescription, wr.regDescription FROM employee e INNER JOIN role r ON
e.roleID = r.roleID INNER JOIN workerRegulation wr ON e.workerRegulationID = wr.workerRegulationID WHERE TIMESTAMPDIFF(YEAR,
STR_TO_DATE(e.startDate, '%d/%m/%Y'), CURDATE()) > 5 AND r.roleDescription IN ('Web Developer', 'Financial Analyst', 'Customer Service
Representative') AND wr.workerRegulationID = '2';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

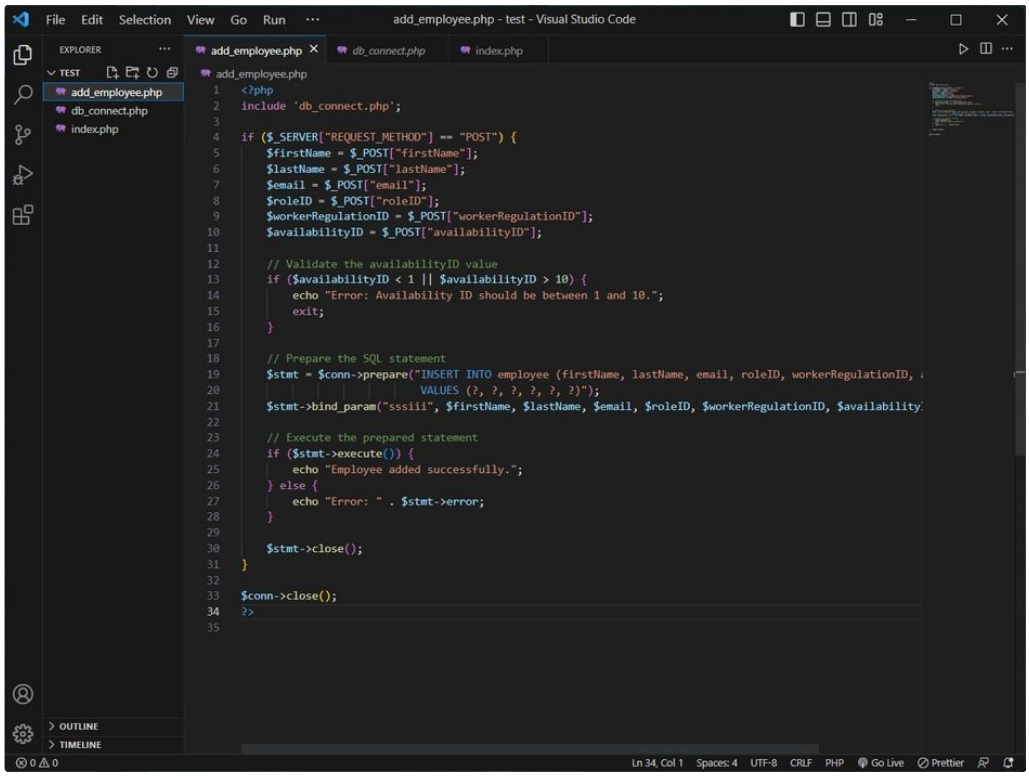
Extra options

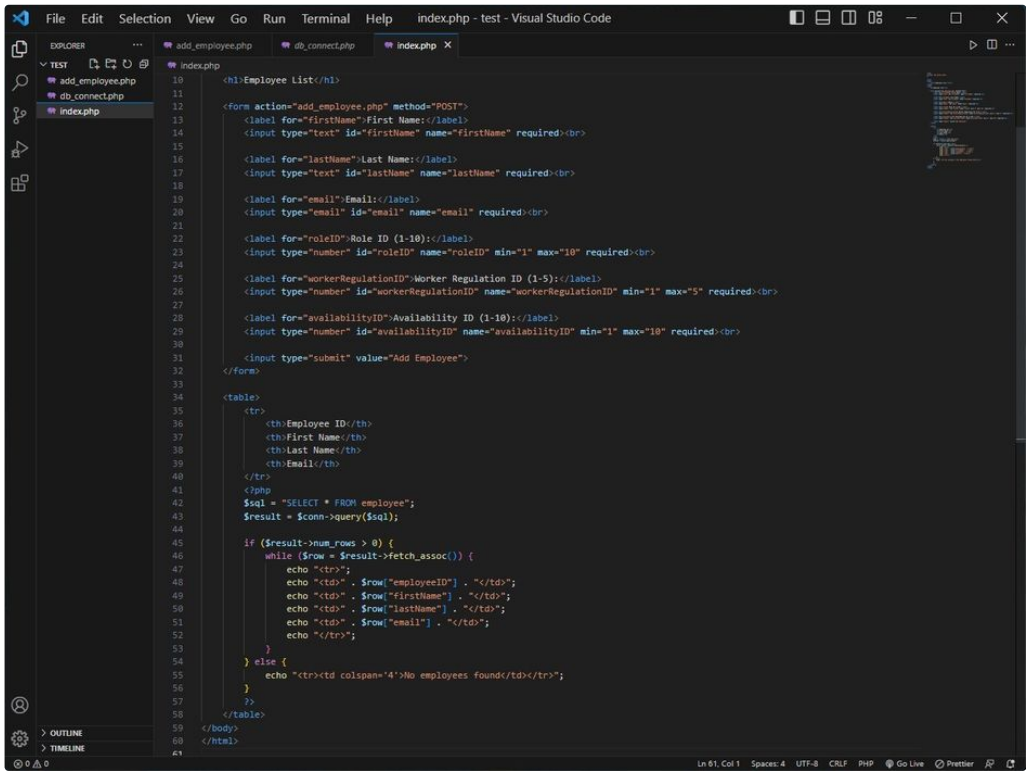
employeeID	firstName	lastName	startDate	roleDescription	regDescription
8	Charles	Martin	17/9/2015	Customer Service Representative	Part Time
11	Tumbleweed	Tumbleweed	1/05/2014	Financial Analyst	Part Time
23	Banana Mania	Banana Mania	7/04/2012	Financial Analyst	Part Time
43	Olivine	Olivine	7/05/2017	Financial Analyst	Part Time
79	Aureolin	Aureolin	4/10/2013	Financial Analyst	Part Time

Major-specific work by Marco Giacoppo

Cybersecurity

Here’s how we used SQL sanitisation using prepared statements:





1. Instead of directly concatenating user input into the SQL query, we use placeholders (?) in the query.
2. Prepare the SQL statement by using the prepare() method of the MySQLi connection object.
3. Bind the user input values to the placeholders using the bind_param() method.
4. Execute the prepared statement using the execute() method.

In this updated code, the prepare() method is used to prepare the SQL statement with placeholders. The bind_param() method is then used to bind the user input values to the placeholders. The "sssiii" argument in the bind_param() method represents the types of the parameters (s for string, i for integer).

Using prepared statements with parameter binding helps protect against SQL injection by automatically handling the escaping of special characters and ensuring the proper interpretation of user input as data rather than executable SQL code.

By implementing SQL sanitisation with prepared statements, we can enhance the security of our application and prevent SQL injection vulnerabilities.

Team Reflection

📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the 4Ls Retrospective Play.

Team	Z
Team members	@Ertesam Zarif @Marco Giacoppo @Puladitha Silva @Astitva Singh
Date	24/05/2023
Retrospective period	February 27 - May 28

4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
• Project Proposal	• Creating a project with new people. • Learning curve with confluence . • Team comradery. • Assigning roles and responsibilities to form a team. • Finding about the client persona and mapping that to an empathy map.	• Finding clarity in what we are supposed to do. • An understanding of databases.	• Not knowing what to do at first. • Learning about databases from the ground up again. • Trying to find the risk assessment for the project.	• How to use Confluence and Jira. • ERD Diagramming.
• Progress Report	• Seeing the database finally come together. • Injecting sample data into the database.	• Weekly team meetings for each topic. • Revisions of the ERD diagram.	• Working on phpMyAdmin for the database.	• Jira road mapping usage.
• Final Product	• Seeing the final product after 12 weeks of work. • Seeing how the group developed over the course of the unit.	• Adding query statements. • How to index.	• Doing the major specific contribution for the database. • Creating the video. • Indexing.	• phpMyAdmin exporting. • Cybersecurity contributions for the database.

⚡ Action plan

Action	Owner	Due date	Action items
ERD Diagram	@Ertesam Zarif	27th April	Finalize the ERD Diagram
Physical implementation	@Marco Giacoppo	4th May	Import the ERD diagram design and create the actual database.
Presentation template	@Puladitha Silva @Astitva Singh	24th May	Create the template and link together the template for the video presentation.
Confluence 4L	@Ertesam Zarif	24th May	Help with the creation of the 4L Retrospective page on confluence.
Major specific work	@Marco Giacoppo	24th May	Do the major specific work including SQL sanitisation