DBMS - Mini Project

PAYROLL MANAGEMENT

Submitted By:

NAME: ANANYA ADIGA
SRN: PES2UG20CS043
V Semester Section A

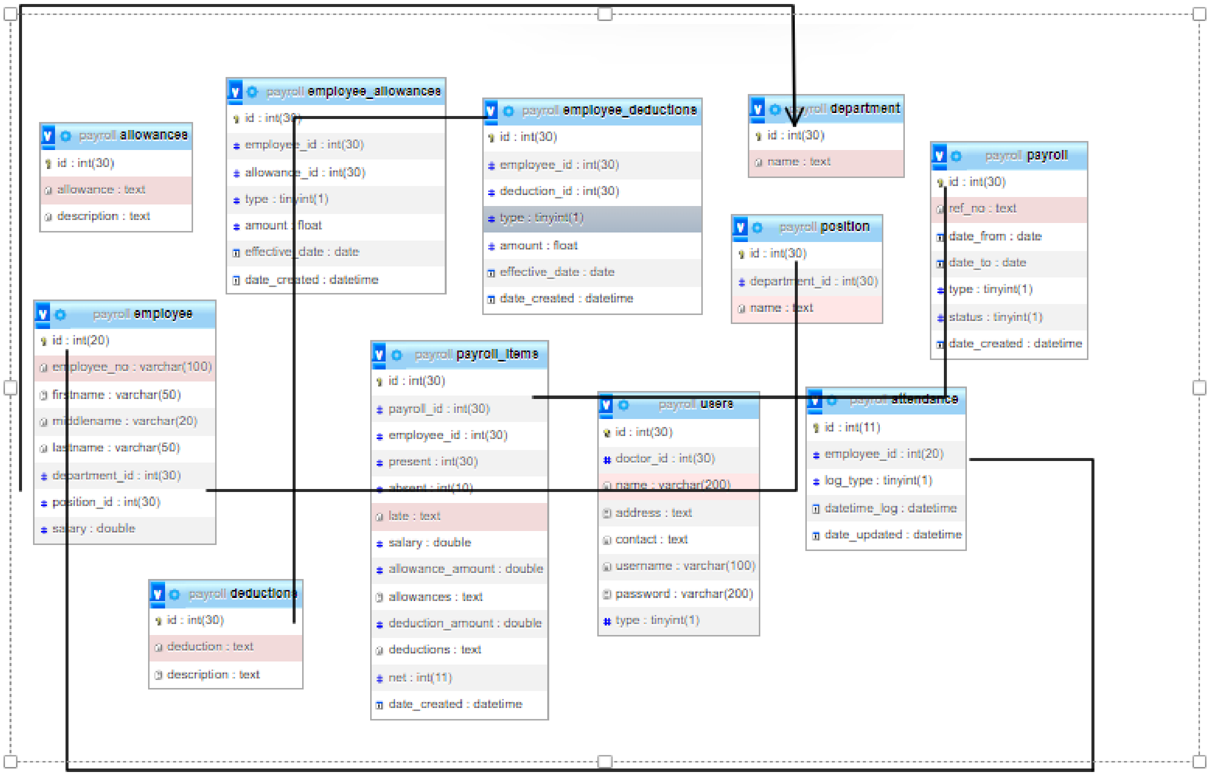
Short Description and Scope of the Project:

A payroll management system is used to manage all the employee’s financial records in an automated manner. It manages employee's salaries, deductions, other conveyance, net pay, bonuses and generation of pay-slips. It includes collecting the list of employees to be paid, tracking the hours worked, calculating the employee's pay, distributing the salary on time, and recording the payroll expense. Employee work times are collected, records and attendance are verified , wages, taxes and adjustments are calculated.

The Admin gets logged in by valid username and password. Admin can add new Employee, add new Department, add new Pay Grade for the employees. Admin can set the ‘from’ and ‘to’ date worked by an employee in a department with specific pay grade. The Admin can generate an automated monthly salary of an employee. The admin can view all the past records of any recorded employee.

ER Diagram

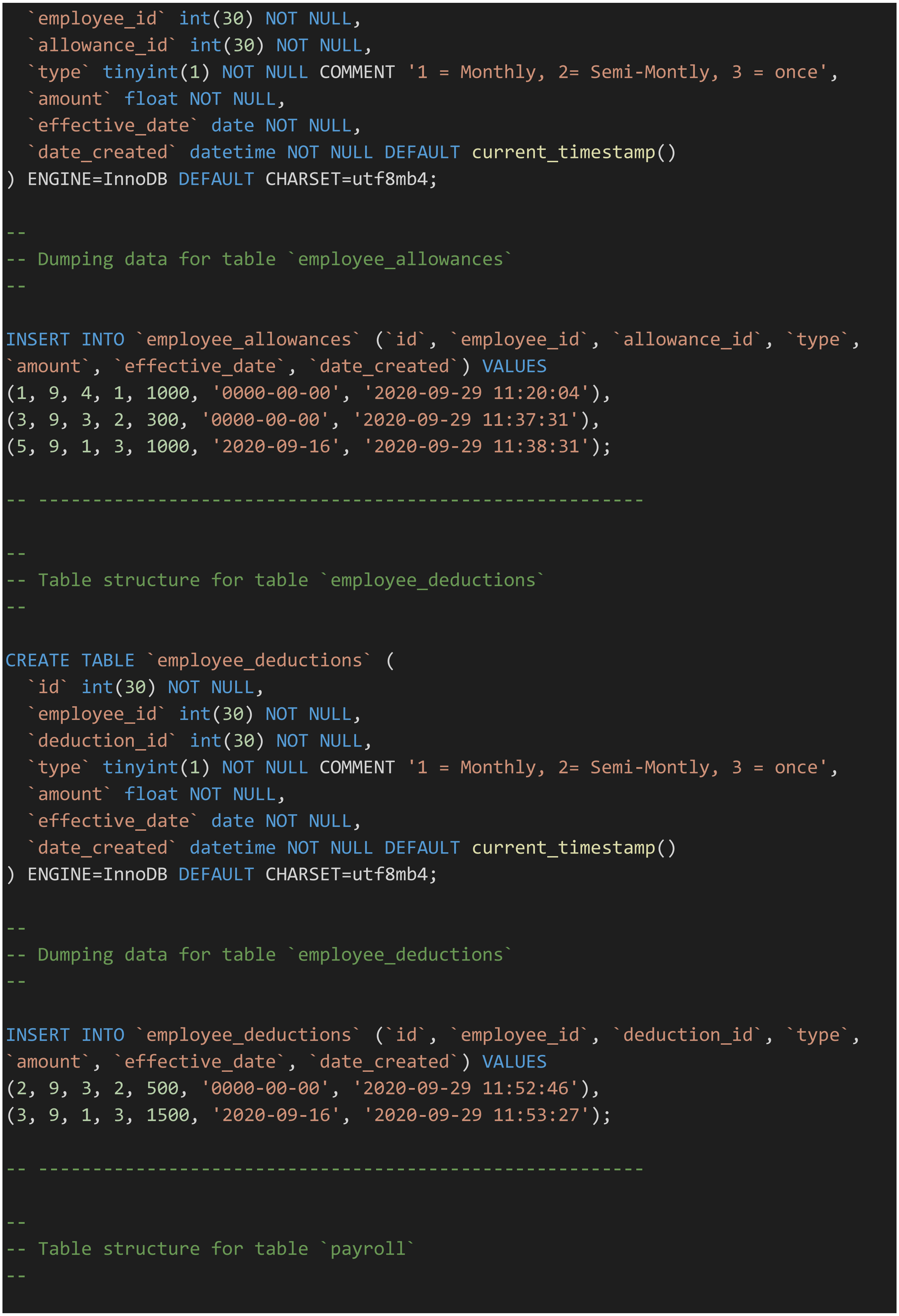
Relational Schema



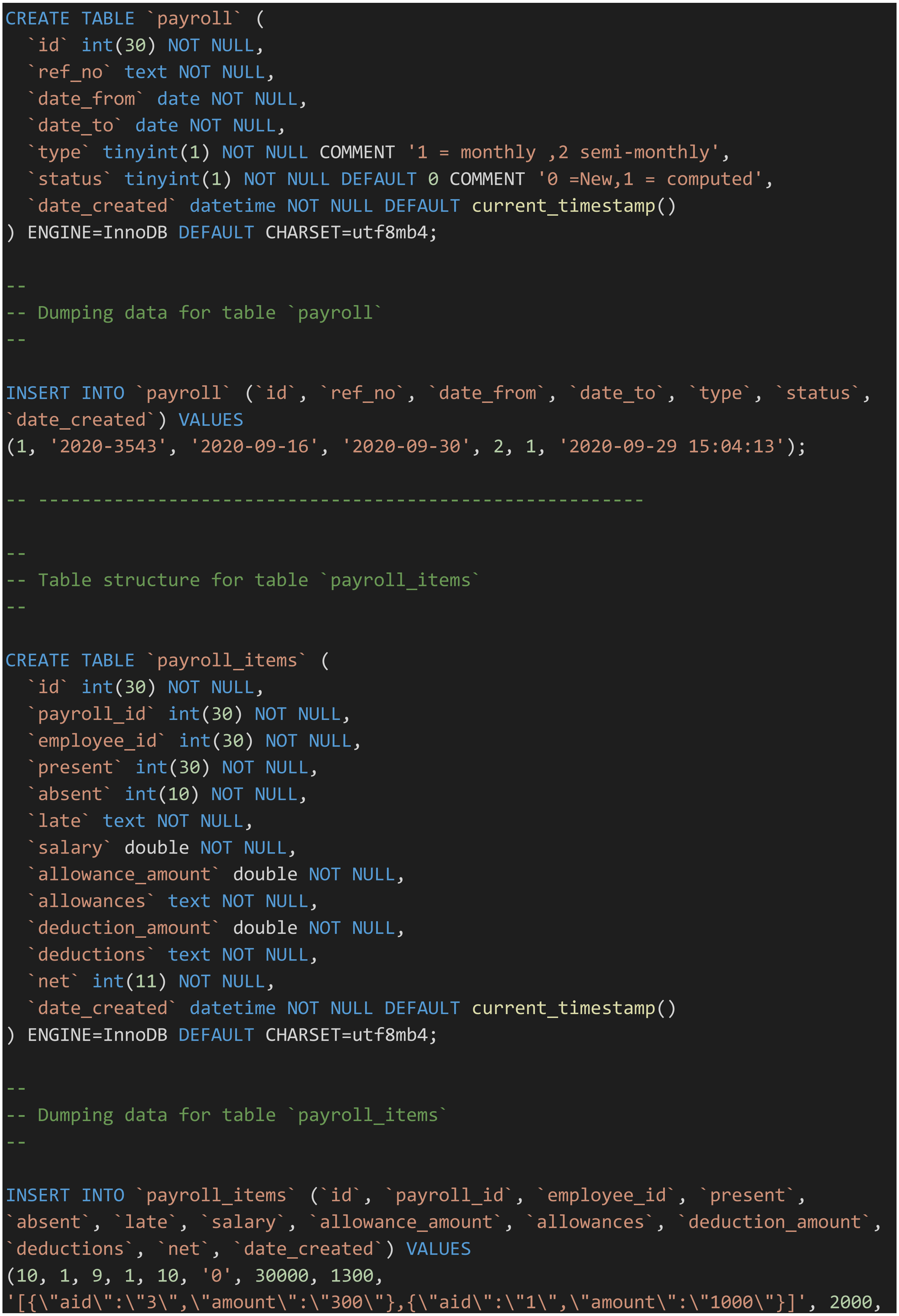
DDL statements - Building the database

```sql
CREATE TABLE `attendance` (
  `id` int(11) NOT NULL,
  `employee_id` int(20) NOT NULL,
  `log_type` tinyint(1) NOT NULL COMMENT '1 = AM IN,2 = AM out, 3= PM IN, 4= PM
out\r\n',
  `datetime_log` datetime NOT NULL DEFAULT current_timestamp(),
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `attendance`
--

INSERT INTO `attendance` (`id`, `employee_id`, `log_type`, `datetime_log`,
`date_updated`) VALUES
(10, 9, 1, '2020-09-16 08:00:00', '2020-09-29 16:16:57'),
(11, 9, 2, '2020-09-16 12:00:00', '2020-09-29 16:16:57'),
(12, 9, 3, '2020-09-16 13:00:00', '2020-09-29 16:16:57'),
(16, 9, 4, '2020-09-16 17:00:00', '2020-09-29 16:16:57');

-- --------------------------------------------------------

--
-- Table structure for table `deductions`
--

CREATE TABLE `deductions` (
  `id` int(30) NOT NULL,
  `deduction` text NOT NULL,
  `description` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `deductions`
--

INSERT INTO `deductions` (`id`, `deduction`, `description`) VALUES
(1, 'Cash Advance', 'Cash Advance'),
(3, 'Sample', 'Sample Deduction');

-- --------------------------------------------------------

--
-- Table structure for table `department`
```

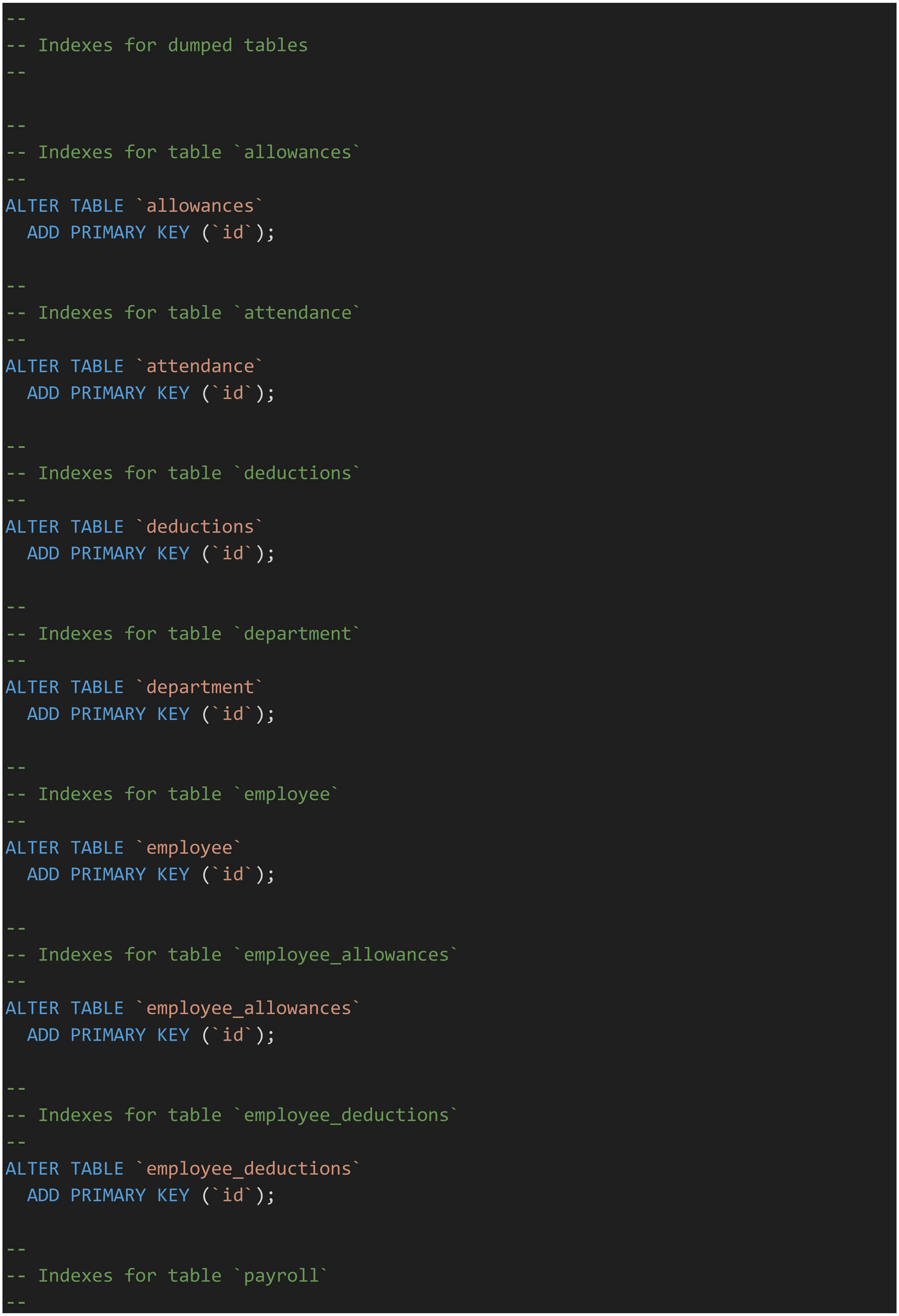
```
--

CREATE TABLE `department` (
  `id` int(30) NOT NULL,
  `name` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `department`
--

INSERT INTO `department` (`id`, `name`) VALUES
(1, 'IT Department'),
(2, 'HR Department'),
(3, 'Accounting and Finance Department');

-- --------------------------------------------------------

--
-- Table structure for table `employee`
--

CREATE TABLE `employee` (
  `id` int(20) NOT NULL,
  `employee_no` varchar(100) NOT NULL,
  `firstname` varchar(50) NOT NULL,
  `middlename` varchar(20) NOT NULL,
  `lastname` varchar(50) NOT NULL,
  `department_id` int(30) NOT NULL,
  `position_id` int(30) NOT NULL,
  `salary` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `employee`
--

INSERT INTO `employee` (`id`, `employee_no`, `firstname`, `middlename`,
`lastname`, `department_id`, `position_id`, `salary`) VALUES
(9, '2020-9838', 'John', 'C', 'Smith', 1, 1, 30000);

-- --------------------------------------------------------

--
-- Table structure for table `employee_allowances`
--

CREATE TABLE `employee_allowances` (
  `id` int(30) NOT NULL,
```



```
  `employee_id` int(30) NOT NULL,
  `allowance_id` int(30) NOT NULL,
  `type` tinyint(1) NOT NULL COMMENT '1 = Monthly, 2= Semi-Montly, 3 = once',
  `amount` float NOT NULL,
  `effective_date` date NOT NULL,
  `date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `employee_allowances`
--

INSERT INTO `employee_allowances` (`id`, `employee_id`, `allowance_id`, `type`,
`amount`, `effective_date`, `date_created`) VALUES
(1, 9, 4, 1, 1000, '0000-00-00', '2020-09-29 11:20:04'),
(3, 9, 3, 2, 300, '0000-00-00', '2020-09-29 11:37:31'),
(5, 9, 1, 3, 1000, '2020-09-16', '2020-09-29 11:38:31');

-- --------------------------------------------------------

--
-- Table structure for table `employee_deductions`
--

CREATE TABLE `employee_deductions` (
  `id` int(30) NOT NULL,
  `employee_id` int(30) NOT NULL,
  `deduction_id` int(30) NOT NULL,
  `type` tinyint(1) NOT NULL COMMENT '1 = Monthly, 2= Semi-Montly, 3 = once',
  `amount` float NOT NULL,
  `effective_date` date NOT NULL,
  `date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `employee_deductions`
--

INSERT INTO `employee_deductions` (`id`, `employee_id`, `deduction_id`, `type`,
`amount`, `effective_date`, `date_created`) VALUES
(2, 9, 3, 2, 500, '0000-00-00', '2020-09-29 11:52:46'),
(3, 9, 1, 3, 1500, '2020-09-16', '2020-09-29 11:53:27');

-- --------------------------------------------------------

--
-- Table structure for table `payroll`
--
```

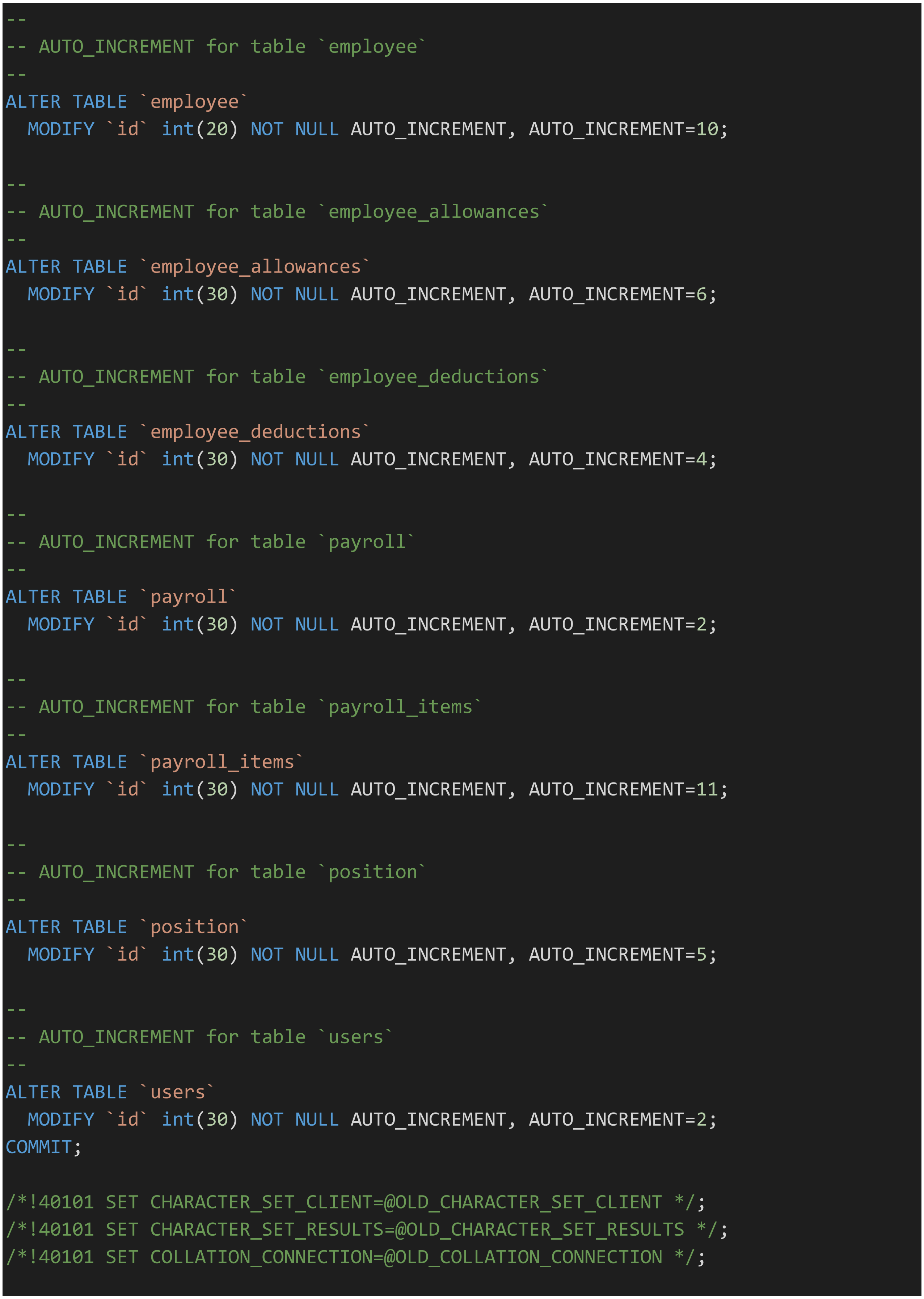


```sql
CREATE TABLE `payroll` (
  `id` int(30) NOT NULL,
  `ref_no` text NOT NULL,
  `date_from` date NOT NULL,
  `date_to` date NOT NULL,
  `type` tinyint(1) NOT NULL COMMENT '1 = monthly ,2 semi-monthly',
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0 =New,1 = computed',
  `date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `payroll`
--

INSERT INTO `payroll` (`id`, `ref_no`, `date_from`, `date_to`, `type`, `status`,
`date_created`) VALUES
(1, '2020-3543', '2020-09-16', '2020-09-30', 2, 1, '2020-09-29 15:04:13');

-- --------------------------------------------------------

--
-- Table structure for table `payroll_items`
--

CREATE TABLE `payroll_items` (
  `id` int(30) NOT NULL,
  `payroll_id` int(30) NOT NULL,
  `employee_id` int(30) NOT NULL,
  `present` int(30) NOT NULL,
  `absent` int(10) NOT NULL,
  `late` text NOT NULL,
  `salary` double NOT NULL,
  `allowance_amount` double NOT NULL,
  `allowances` text NOT NULL,
  `deduction_amount` double NOT NULL,
  `deductions` text NOT NULL,
  `net` int(11) NOT NULL,
  `date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `payroll_items`
--

INSERT INTO `payroll_items` (`id`, `payroll_id`, `employee_id`, `present`,
`absent`, `late`, `salary`, `allowance_amount`, `allowances`, `deduction_amount`,
`deductions`, `net`, `date_created`) VALUES
(10, 1, 9, 1, 10, '0', 30000, 1300,
'[{\"aid\":\"3\",\"amount\":\"300\"},{\"aid\":\"1\",\"amount\":\"1000\"}]', 2000,
'[{\"did\":\"3\",\"amount\":\"500\"},{\"did\":\"1\",\"amount\":\"1500\"}]', 664,
'2020-09-29 18:46:59');

-- --------------------------------------------------------

--
-- Table structure for table `position`
--

CREATE TABLE `position` (
  `id` int(30) NOT NULL,
  `department_id` int(30) NOT NULL,
  `name` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `position`
--

INSERT INTO `position` (`id`, `department_id`, `name`) VALUES
(1, 1, 'Programmer'),
(2, 2, 'HR Supervisor'),
(4, 3, 'Accounting Clerk');

-- --------------------------------------------------------

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(30) NOT NULL,
  `doctor_id` int(30) NOT NULL,
  `name` varchar(200) NOT NULL,
  `address` text NOT NULL,
  `contact` text NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(200) NOT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 2 COMMENT '1=admin , 2 = staff'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `doctor_id`, `name`, `address`, `contact`, `username`,
`password`, `type`) VALUES
(1, 0, 'Administrator', '', '', 'admin', 'admin123', 1);
```



```sql
--
-- Indexes for dumped tables
--

--
-- Indexes for table `allowances`
--
ALTER TABLE `allowances`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `attendance`
--
ALTER TABLE `attendance`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `deductions`
--
ALTER TABLE `deductions`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `department`
--
ALTER TABLE `department`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `employee`
--
ALTER TABLE `employee`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `employee_allowances`
--
ALTER TABLE `employee_allowances`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `employee_deductions`
--
ALTER TABLE `employee_deductions`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `payroll`
--
```

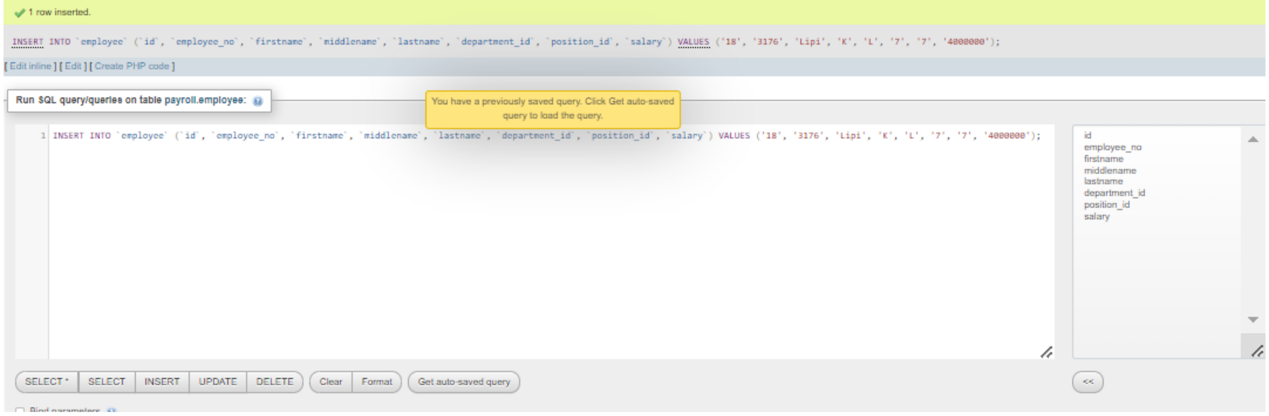
```sql
ALTER TABLE `payroll`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `payroll_items`
--
ALTER TABLE `payroll_items`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `position`
--
ALTER TABLE `position`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `allowances`
--
ALTER TABLE `allowances`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `attendance`
--
ALTER TABLE `attendance`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;

--
-- AUTO_INCREMENT for table `deductions`
--
ALTER TABLE `deductions`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `department`
--
ALTER TABLE `department`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
```



```
--
-- AUTO_INCREMENT for table `employee`
--
ALTER TABLE `employee`
  MODIFY `id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;

--
-- AUTO_INCREMENT for table `employee_allowances`
--
ALTER TABLE `employee_allowances`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `employee_deductions`
--
ALTER TABLE `employee_deductions`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `payroll`
--
ALTER TABLE `payroll`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT for table `payroll_items`
--
ALTER TABLE `payroll_items`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

--
-- AUTO_INCREMENT for table `position`
--
ALTER TABLE `position`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Populating the Database

Inserting into employee table



✓ 1 row inserted.

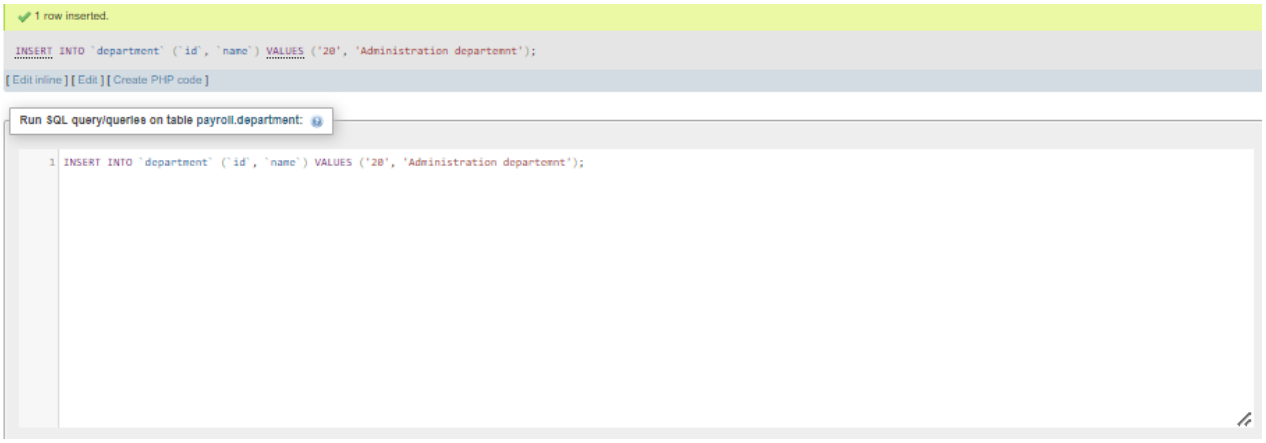
```
INSERT INTO `employee` (`id`, `employee_no`, `firstname`, `middlename`, `lastname`, `department_id`, `position_id`, `salary`) VALUES ('18', '3176', 'Lipi', 'K', 'L', '7', '7', '4000000');
```

[Edit inline] [Edit] [Create PHP code]
Run SQL query/queries on table payroll.employee:
You have a previously saved query. Click Get auto-saved query to load the query.

```
INSERT INTO `employee` (`id`, `employee_no`, `firstname`, `middlename`, `lastname`, `department_id`, `position_id`, `salary`) VALUES ('18', '3176', 'Lipi', 'K', 'L', '7', '7', '4000000');
```

id
employee_no
firstname
middlename
lastname
department_id
position_id
salary
SELECT *
SELECT
INSERT
UPDATE
DELETE
Clear
Format
Get auto-saved query
<<
Bind parameters

Inserting into department table:



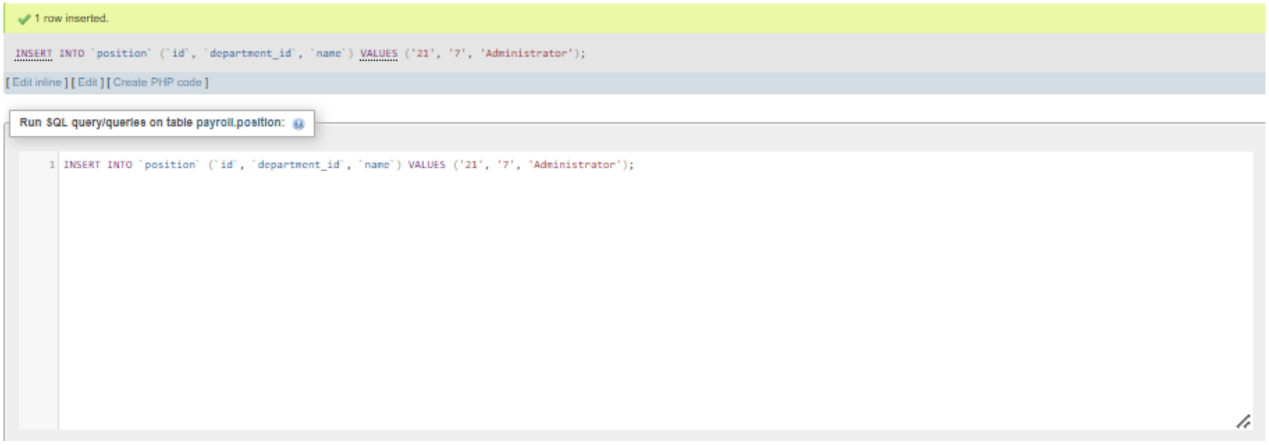
✓ 1 row inserted.

```
INSERT INTO `department` (`id`, `name`) VALUES ('20', 'Administration departemnt');
```

[Edit inline] [Edit] [Create PHP code]
Run SQL query/queries on table payroll.department:

```
INSERT INTO `department` (`id`, `name`) VALUES ('20', 'Administration departemnt');
```

Inserting into position table:



✓ 1 row inserted.

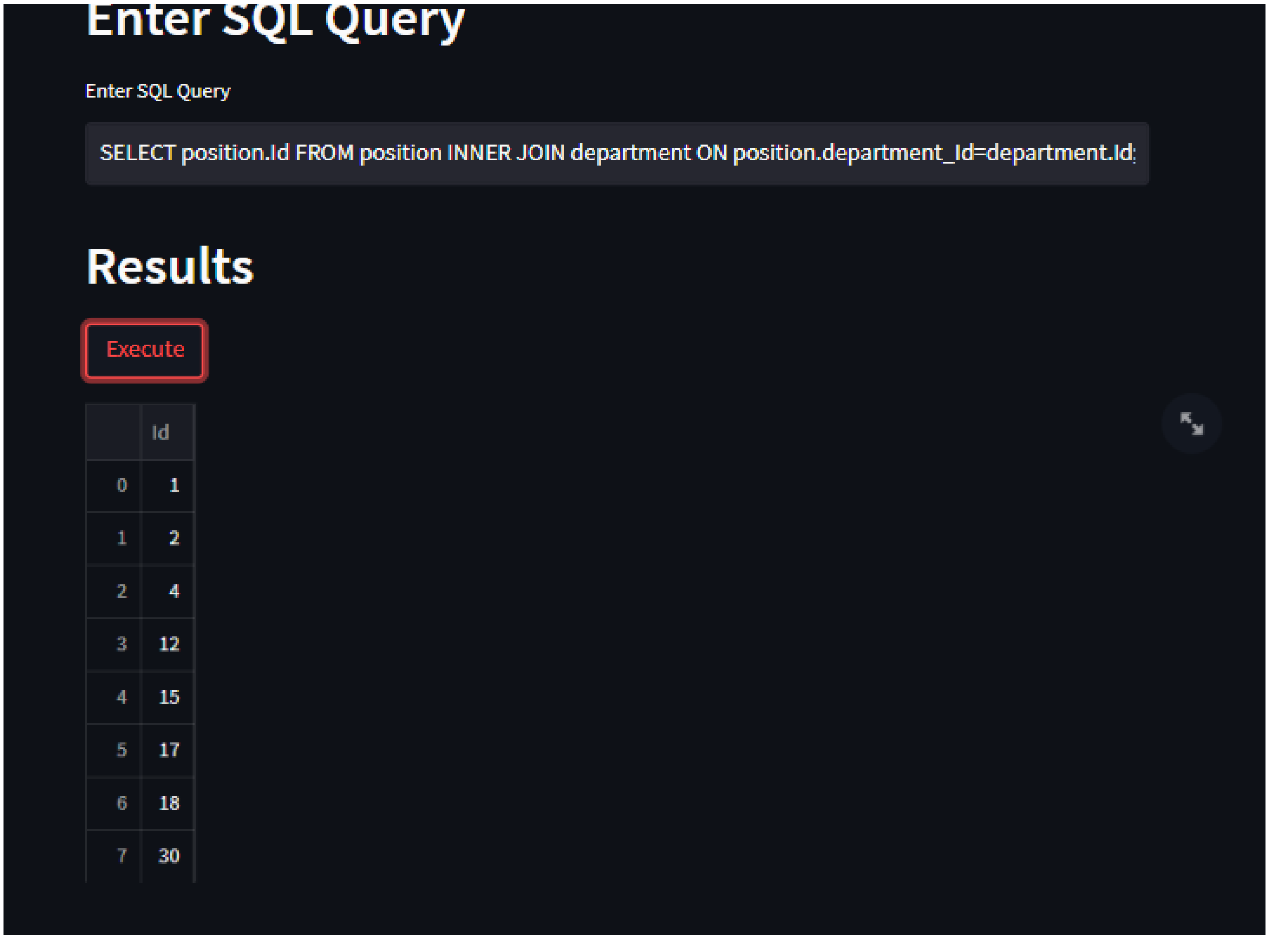
```
INSERT INTO `position` (`id`, `department_id`, `name`) VALUES ('21', '7', 'Administrator');
```

[Edit inline] [Edit] [Create PHP code]
Run SQL query/queries on table payroll.position:

```
INSERT INTO `position` (`id`, `department_id`, `name`) VALUES ('21', '7', 'Administrator');
```

Join Queries

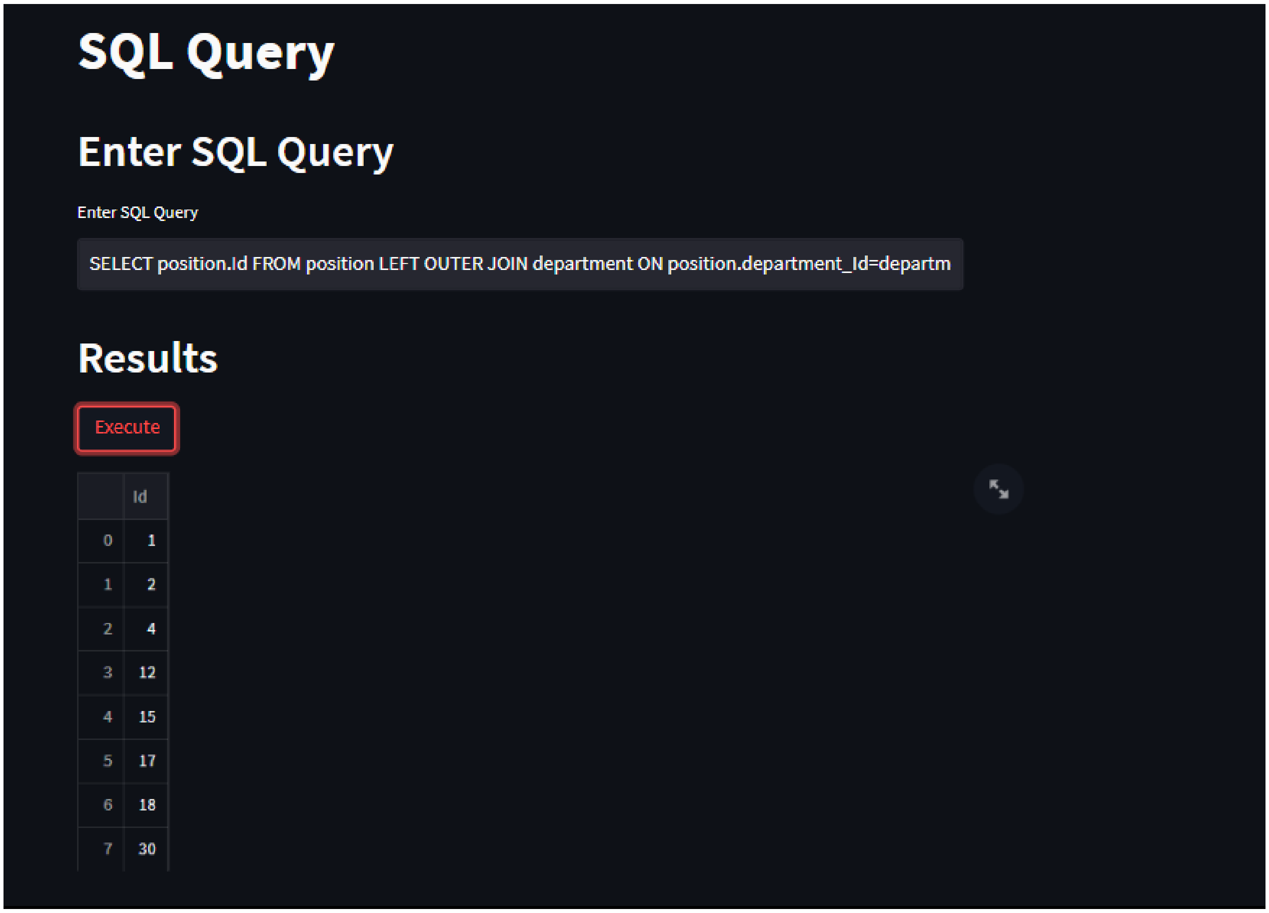
INNER JOIN: joining the id from department and position tables and display the id from position table
SELECT position.Id FROM position INNER JOIN department ON department_Id-department.Id;



SELECT position.Id FROM position INNER JOIN department ON position.department_Id=department.Id;

	Id
0	1
1	2
2	4
3	12
4	15
5	17
6	18
7	30

LEFT OUTER JOIN:
Performing left outer join on position and department table and display the id from position table



SQL Query

Enter SQL Query

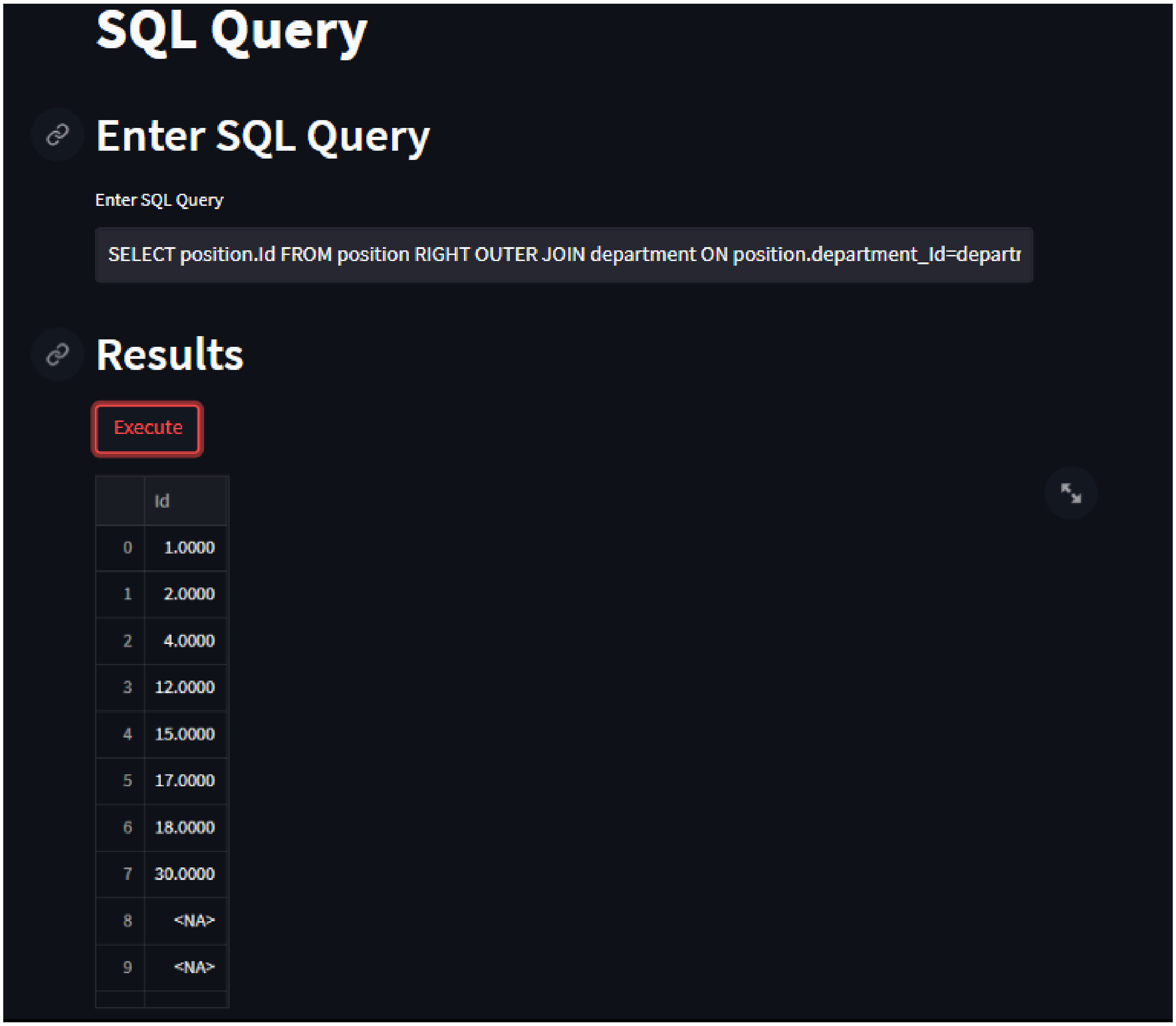
Enter SQL Query

```
SELECT position.Id FROM position LEFT OUTER JOIN department ON position.department_Id=departm
```

Results

	Id
0	1
1	2
2	4
3	12
4	15
5	17
6	18
7	30

RIGHT OUTER JOIN:



SQL Query

Enter SQL Query

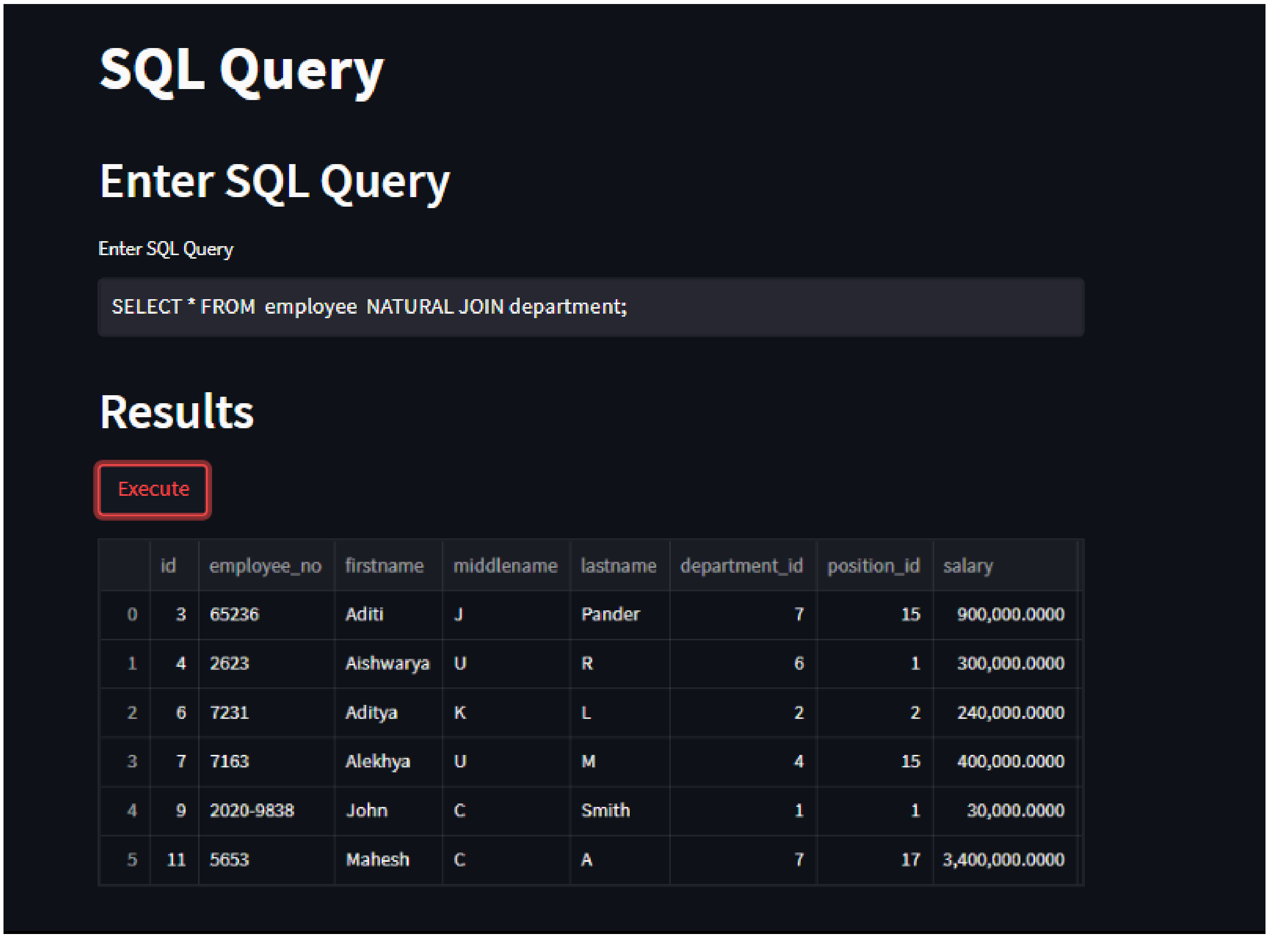
Enter SQL Query

```
SELECT position.Id FROM position RIGHT OUTER JOIN department ON position.department_Id=departr
```

Results

	Id
0	1.0000
1	2.0000
2	4.0000
3	12.0000
4	15.0000
5	17.0000
6	18.0000
7	30.0000
8	<NA>
9	<NA>

NATURAL JOIN:



SQL Query

Enter SQL Query

Enter SQL Query

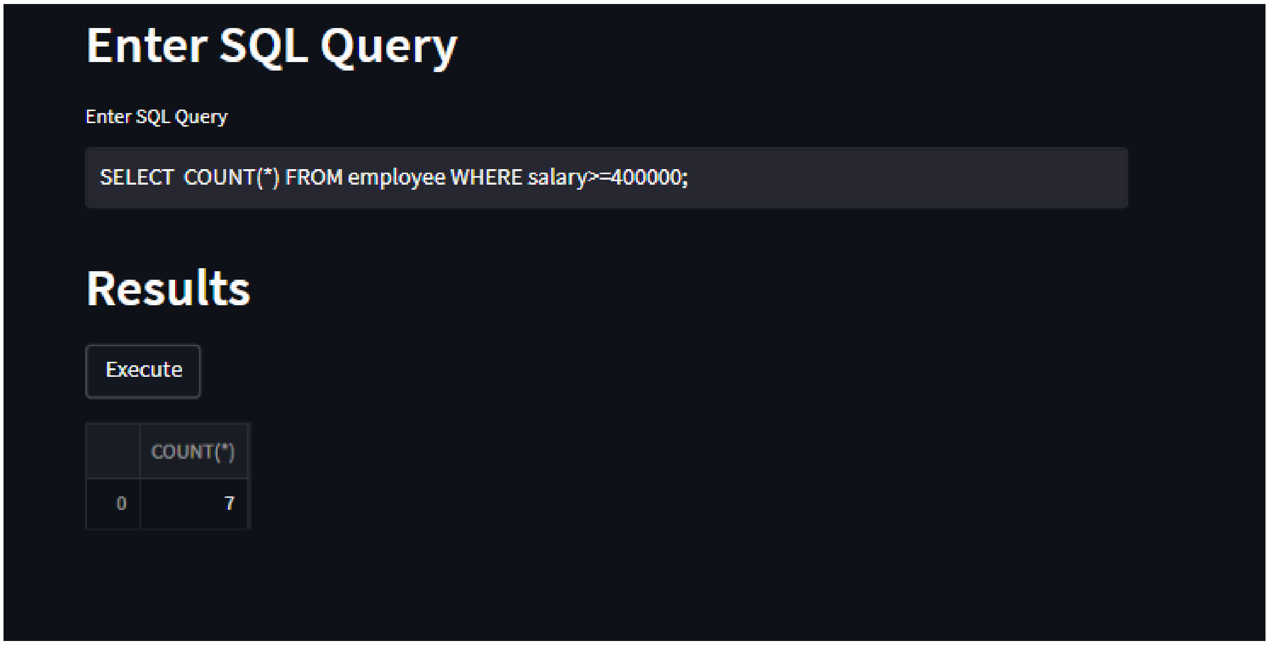
```
SELECT * FROM  employee  NATURAL JOIN department;
```

Results

	id	employee_no	firstname	middlename	lastname	department_id	position_id	salary
0	3	65236	Aditi	J	Pander	7	15	900,000.0000
1	4	2623	Aishwarya	U	R	6	1	300,000.0000
2	6	7231	Aditya	K	L	2	2	240,000.0000
3	7	7163	Alekhya	U	M	4	15	400,000.0000
4	9	2020-9838	John	C	Smith	1	1	30,000.0000
5	11	5653	Mahesh	C	A	7	17	3,400,000.0000

Aggregate Functions

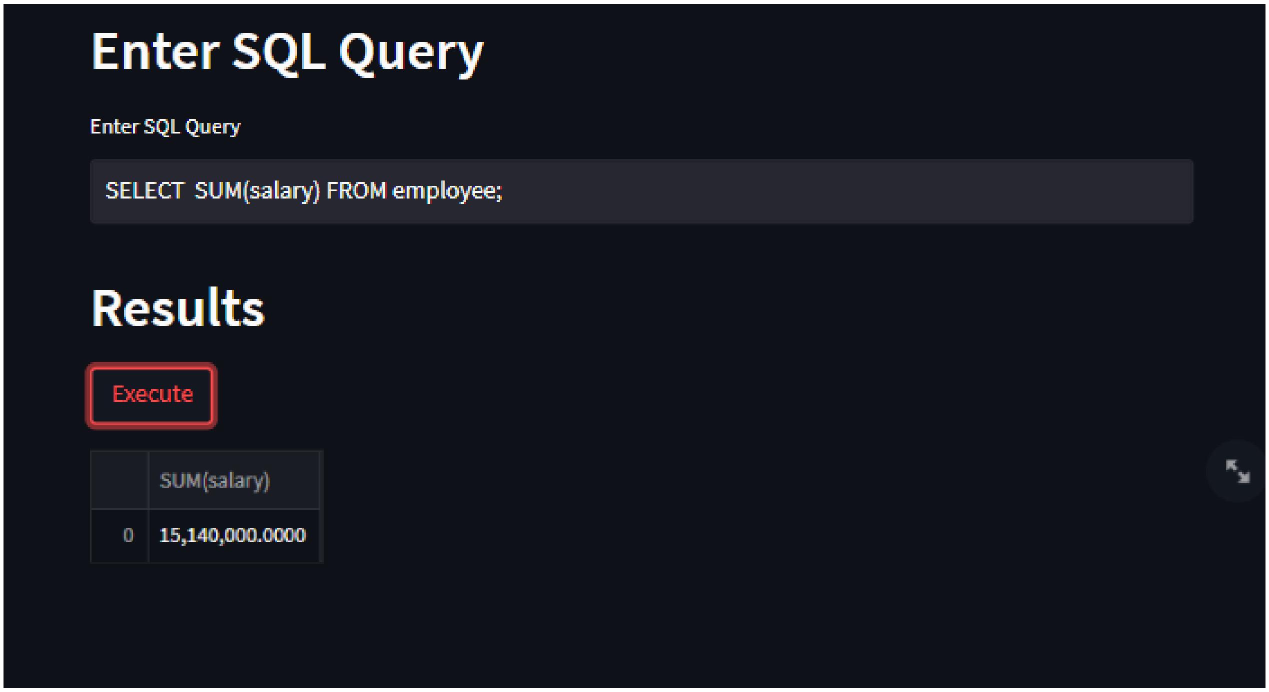
COUNT- number of employees with salary more than 400000



```sql
SELECT  COUNT(*) FROM employee WHERE salary>=400000;
```

	COUNT(*)
0	7

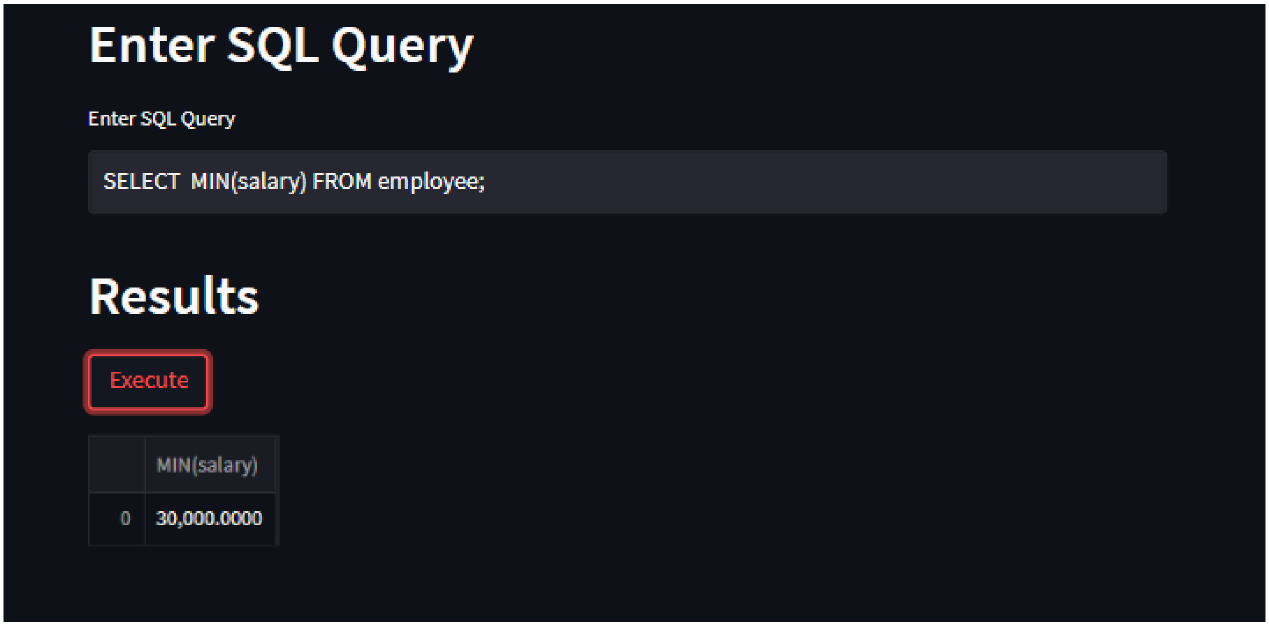
SUM- total salary given to all the employees



```sql
SELECT  SUM(salary) FROM employee;
```

	SUM(salary)
0	15,140,000.0000

MIN- find the least salary



Enter SQL Query

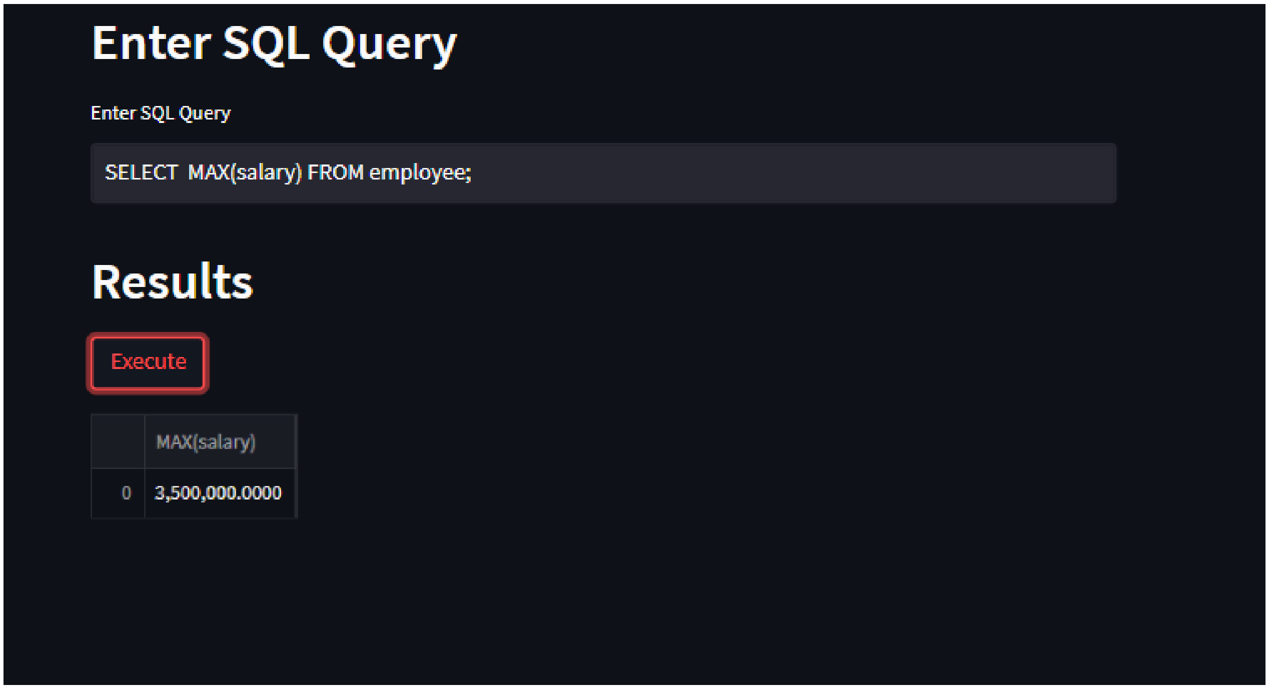
Enter SQL Query

```
SELECT  MIN(salary) FROM employee;
```

Results

	MIN(salary)
0	30,000.0000

MAX- find the highest salary



Enter SQL Query

Enter SQL Query

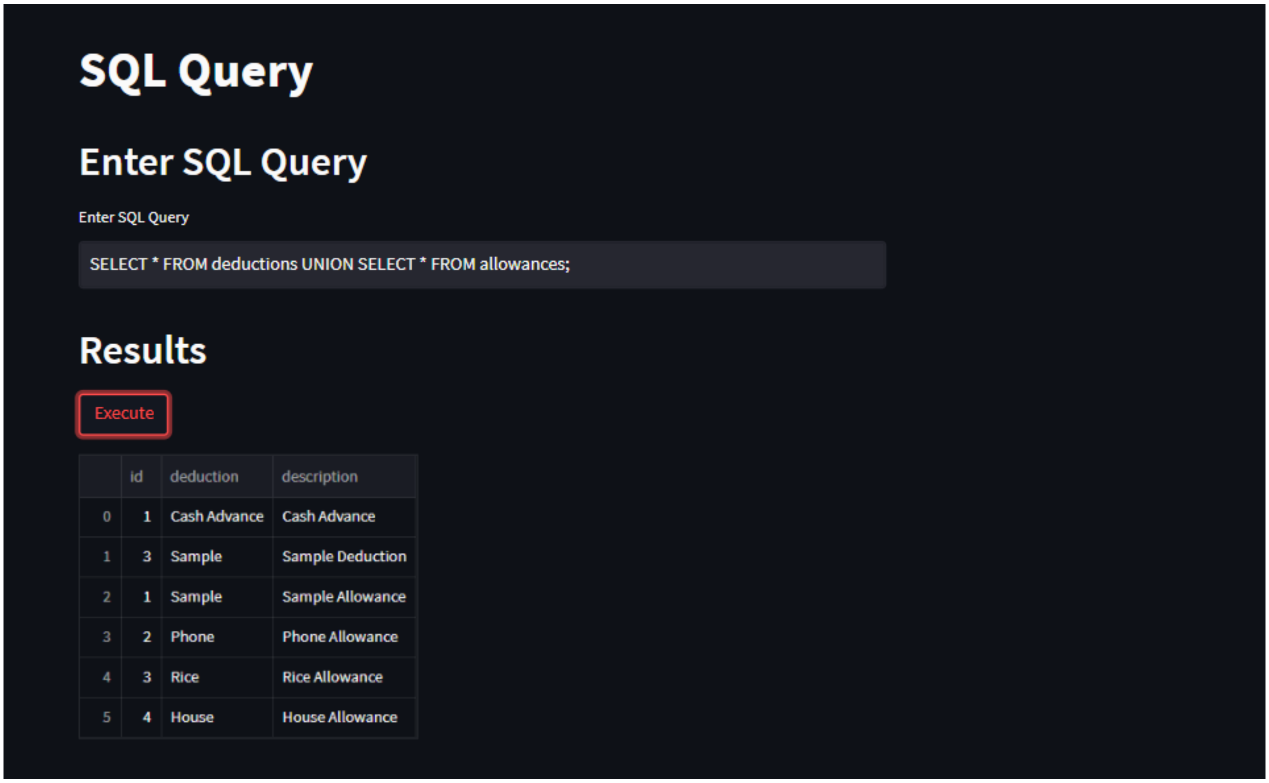
```
SELECT  MAX(salary) FROM employee;
```

Results

	MAX(salary)
0	3,500,000.0000

Set Operations

UNION- find the union of deductions and allowances tables



SQL Query

Enter SQL Query

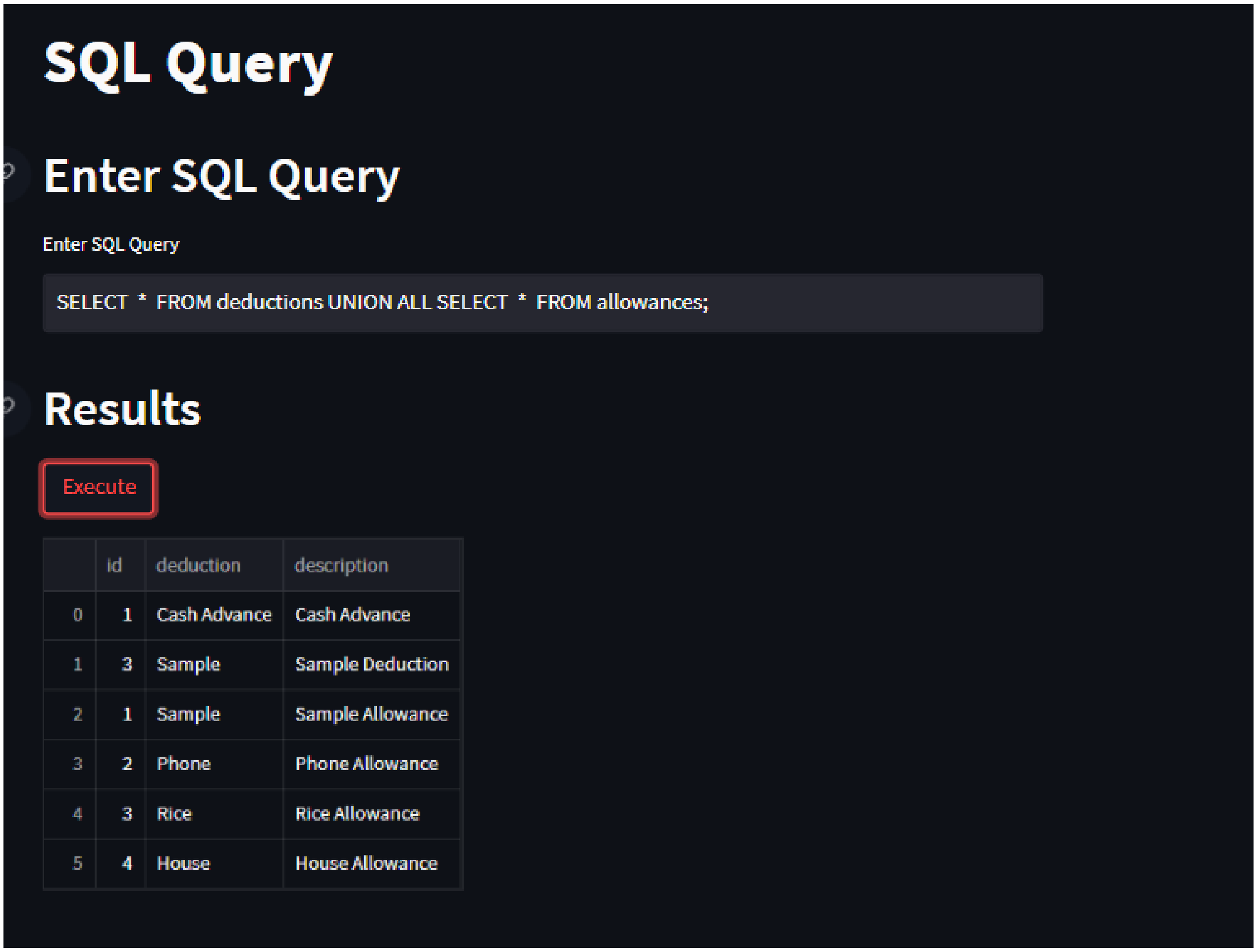
Enter SQL Query

```
SELECT * FROM deductions UNION SELECT * FROM allowances;
```

Results

	id	deduction	description
0	1	Cash Advance	Cash Advance
1	3	Sample	Sample Deduction
2	1	Sample	Sample Allowance
3	2	Phone	Phone Allowance
4	3	Rice	Rice Allowance
5	4	House	House Allowance

UNION ALL- find the union of all rows in deductions and allowances tables



SQL Query

Enter SQL Query

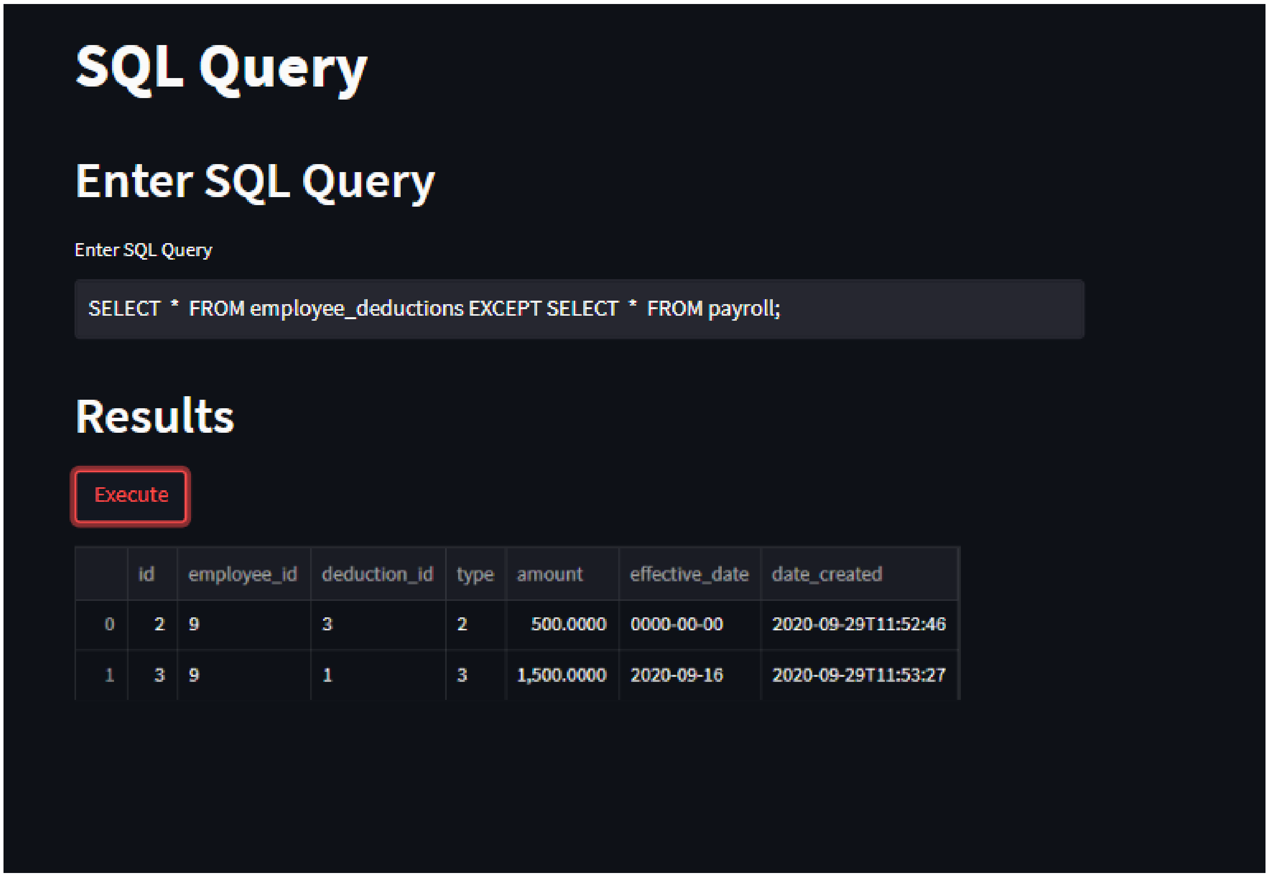
Enter SQL Query

```
SELECT * FROM deductions UNION ALL SELECT * FROM allowances;
```

Results

	id	deduction	description
0	1	Cash Advance	Cash Advance
1	3	Sample	Sample Deduction
2	1	Sample	Sample Allowance
3	2	Phone	Phone Allowance
4	3	Rice	Rice Allowance
5	4	House	House Allowance

EXCEPT- find all the instances that exist in employee deductions table and not in payroll



SQL Query

Enter SQL Query

Enter SQL Query

```
SELECT * FROM employee_deductions EXCEPT SELECT * FROM payroll;
```

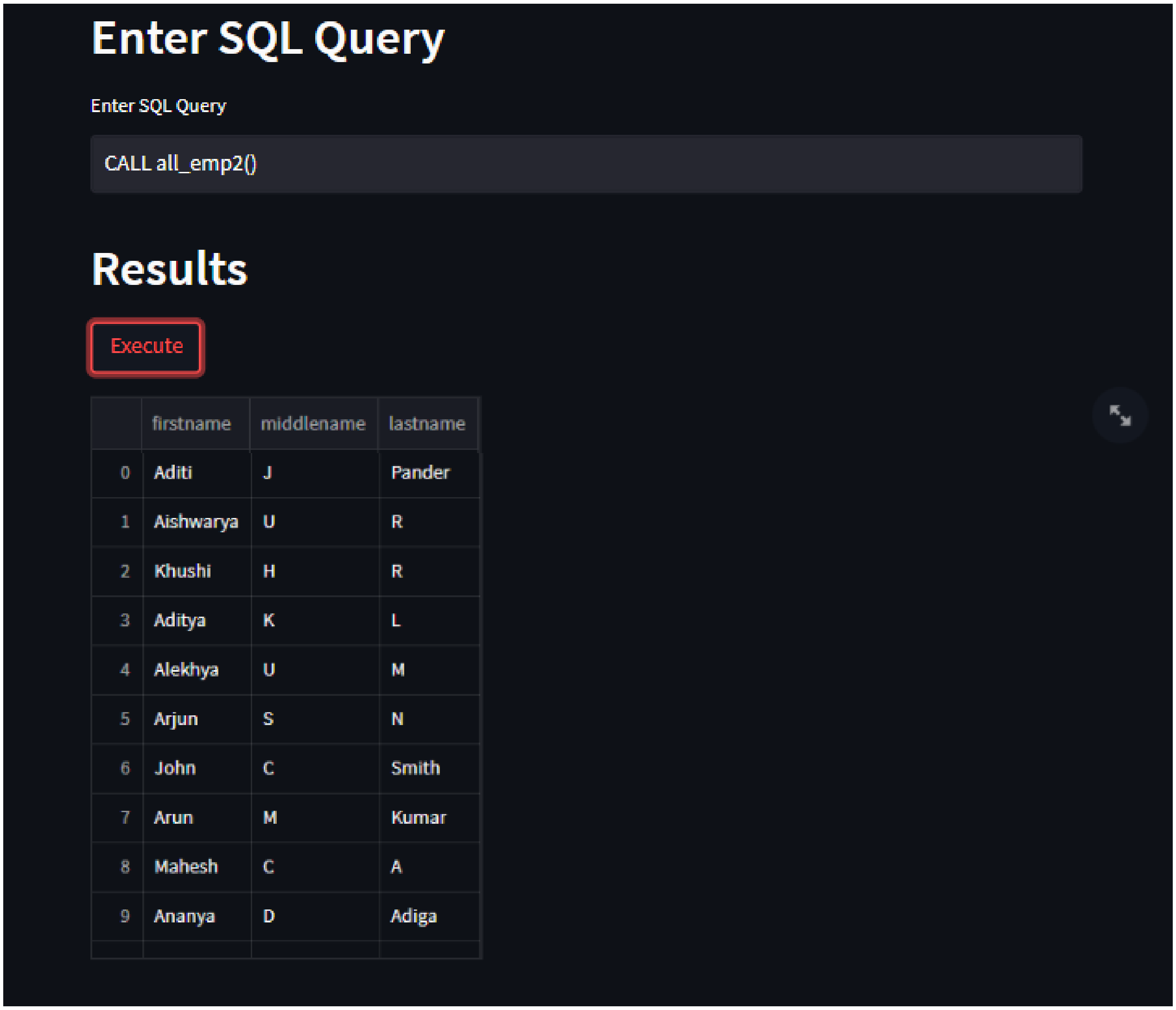
Results

	id	employee_id	deduction_id	type	amount	effective_date	date_created
0	2	9	3	2	500.0000	0000-00-00	2020-09-29T11:52:46
1	3	9	1	3	1,500.0000	2020-09-16	2020-09-29T11:53:27

Functions and Procedures

Procedure to display first, middle and last name of employee

CREATE PROCEDURE all_emp3() BEGIN SELECT firstname, middlename, lastname FROM employee; END



Enter SQL Query

Enter SQL Query

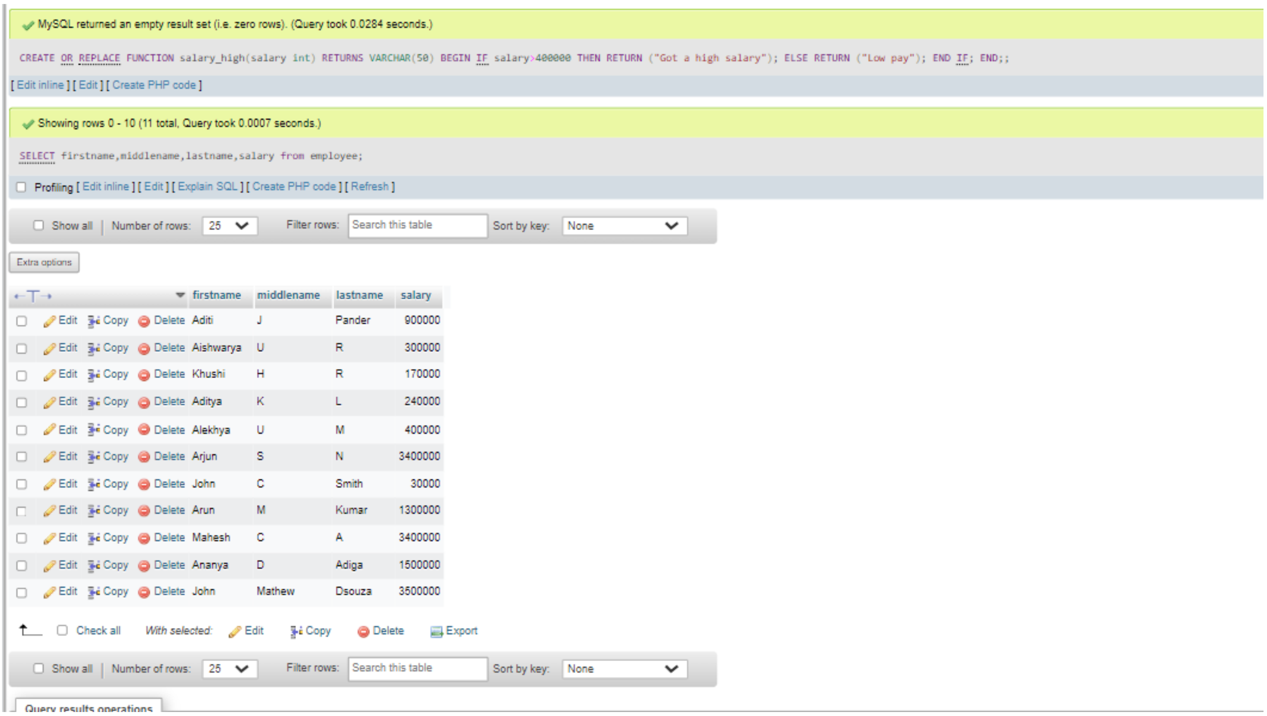
CALL all_emp2()

Results

	firstname	middlename	lastname
0	Aditi	J	Pander
1	Aishwarya	U	R
2	Khushi	H	R
3	Aditya	K	L
4	Alekhya	U	M
5	Arjun	S	N
6	John	C	Smith
7	Arun	M	Kumar
8	Mahesh	C	A
9	Ananya	D	Adiga

FUNCTIONS- to display the employees who have higher pay and lower pay

```
CREATE OR REPLACE FUNCTION salary_high(salary int) RETURNS VARCHAR(50) BEGIN IF salary>400000 THEN RETURN ("Got a high salary"); ELSE RETURN ("Low pay"); END IF; END;;
```



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0284 seconds.)

```
CREATE OR REPLACE FUNCTION salary_high(salary int) RETURNS VARCHAR(50) BEGIN IF salary>400000 THEN RETURN ("Got a high salary"); ELSE RETURN ("Low pay"); END IF; END;;
```

[Edit inline] [Edit] [Create PHP code]

✓ Showing rows 0 - 10 (11 total, Query took 0.0007 seconds.)

```
SELECT firstname,middlename,lastname,salary from employee;
```

□ 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

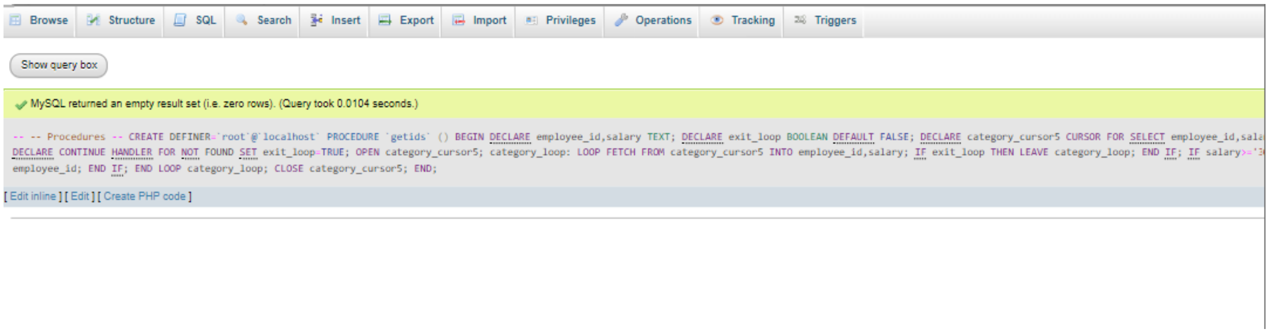
←T→	firstname	middlename	lastname	salary
□ Edit Copy Delete	Aditi	J	Pander	900000
□ Edit Copy Delete	Aishwarya	U	R	300000
□ Edit Copy Delete	Khushi	H	R	170000
□ Edit Copy Delete	Aditya	K	L	240000
□ Edit Copy Delete	Alekhya	U	M	400000
□ Edit Copy Delete	Arjun	S	N	3400000
□ Edit Copy Delete	John	C	Smith	30000
□ Edit Copy Delete	Arun	M	Kumar	1300000
□ Edit Copy Delete	Mahesh	C	A	3400000
□ Edit Copy Delete	Ananya	D	Adiga	1500000
□ Edit Copy Delete	John	Mathew	Dsouza	3500000

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

Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

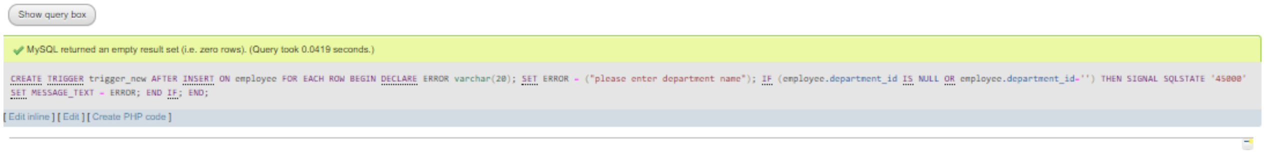
CURSOR- fetch the rows of employees who have salary more than 400000



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0104 seconds.)

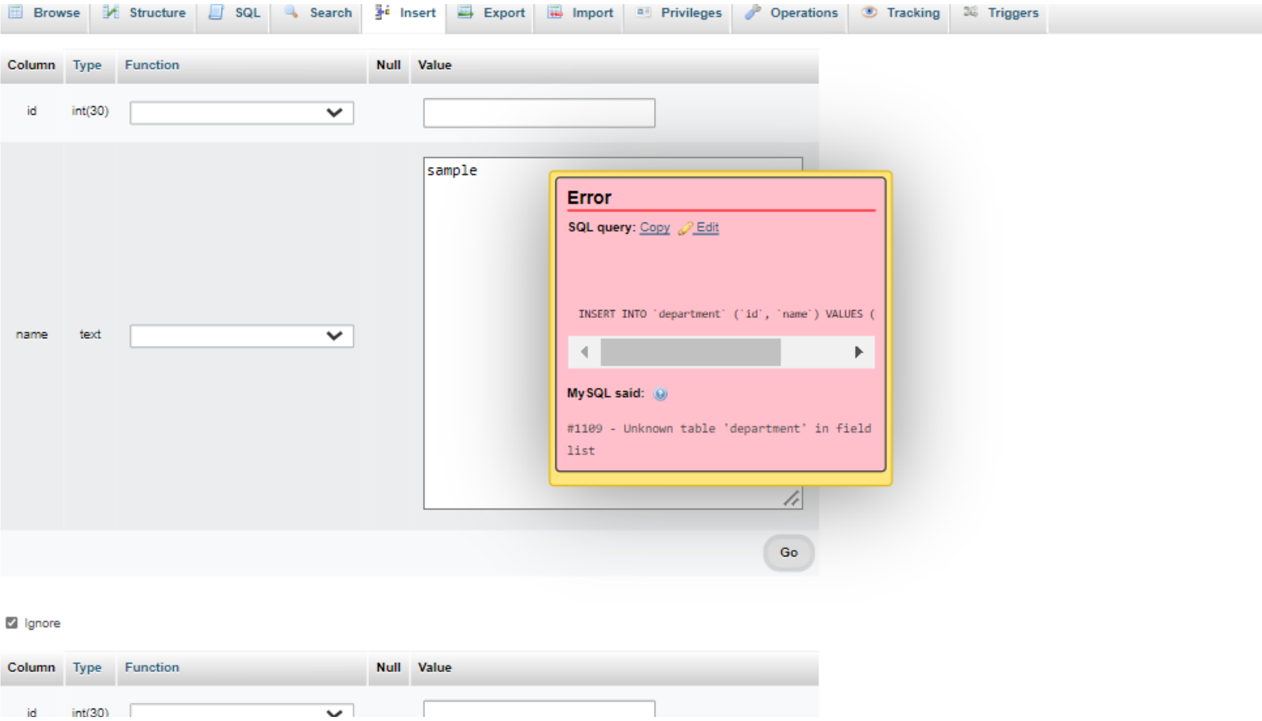
```
-- -- Procedures -- CREATE DEFINER=`root`@`localhost` PROCEDURE `getids` () BEGIN DECLARE employee_id,salary TEXT; DECLARE exit_loop BOOLEAN DEFAULT FALSE; DECLARE category_cursor5 CURSOR FOR SELECT employee_id,sala
DECLARE CONTINUE HANDLER FOR NOT FOUND SET exit_loop=TRUE; OPEN category_cursor5; category_loop: LOOP FETCH FROM category_cursor5 INTO employee_id,salary; IF exit_loop THEN LEAVE category_loop; END IF; IF salary>='3
employee_id; END IF; END LOOP category_loop; CLOSE category_cursor5; END;
```

TRIGGER- display an error message if the department id is not given while inserting rows into department table



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0419 seconds.)

```
CREATE TRIGGER trigger_new AFTER INSERT ON employee FOR EACH ROW BEGIN DECLARE ERROR varchar(20); SET ERROR = ("please enter department name"); IF (employee.department_id IS NULL OR employee.department_id='') THEN SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = ERROR; END IF; END;
```



Column	Type	Function	Null	Value
id	int(30)			
name	text			sample

Error

SQL query: Copy Edit

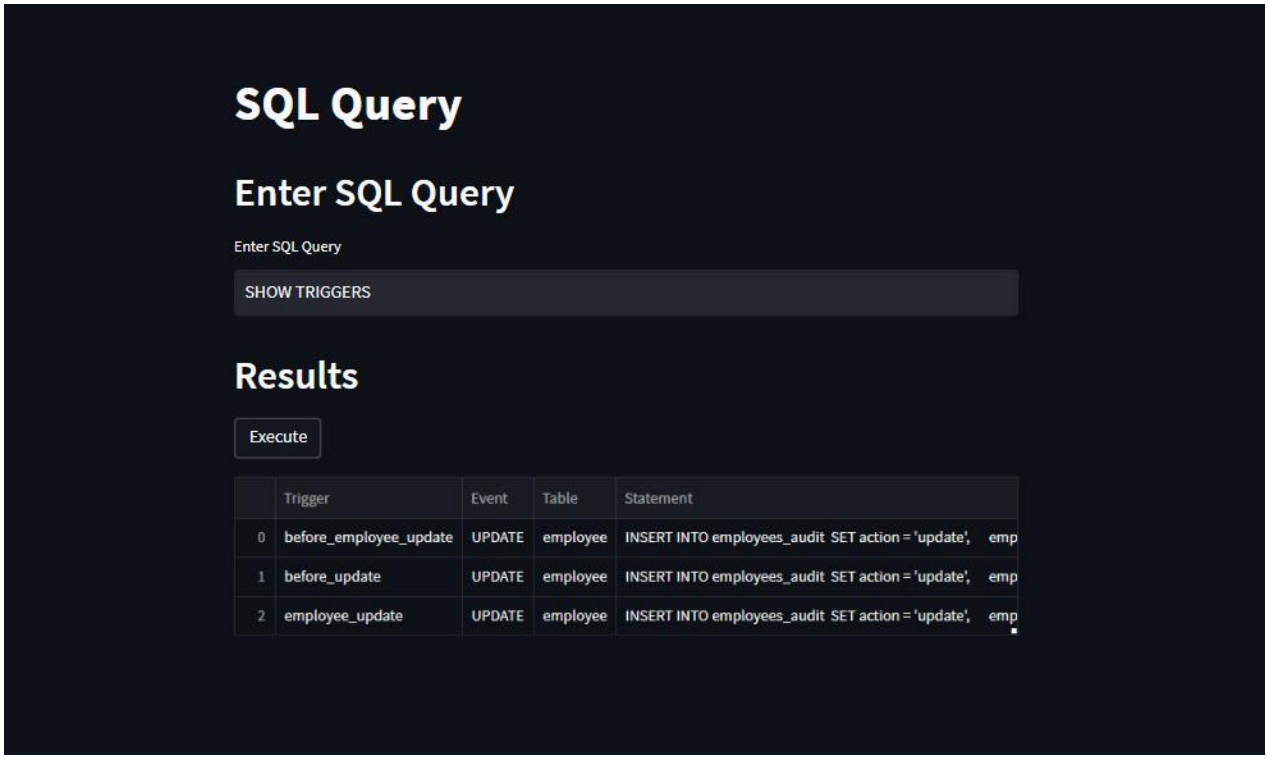
```
INSERT INTO `department` (`id`, `name`) VALUES (
```

MySQL said:

```
#1109 - Unknown table 'department' in field list
```

☑ Ignore

Column	Type	Function	Null	Value
id	int(30)			



SQL Query

Enter SQL Query

Enter SQL Query

SHOW TRIGGERS

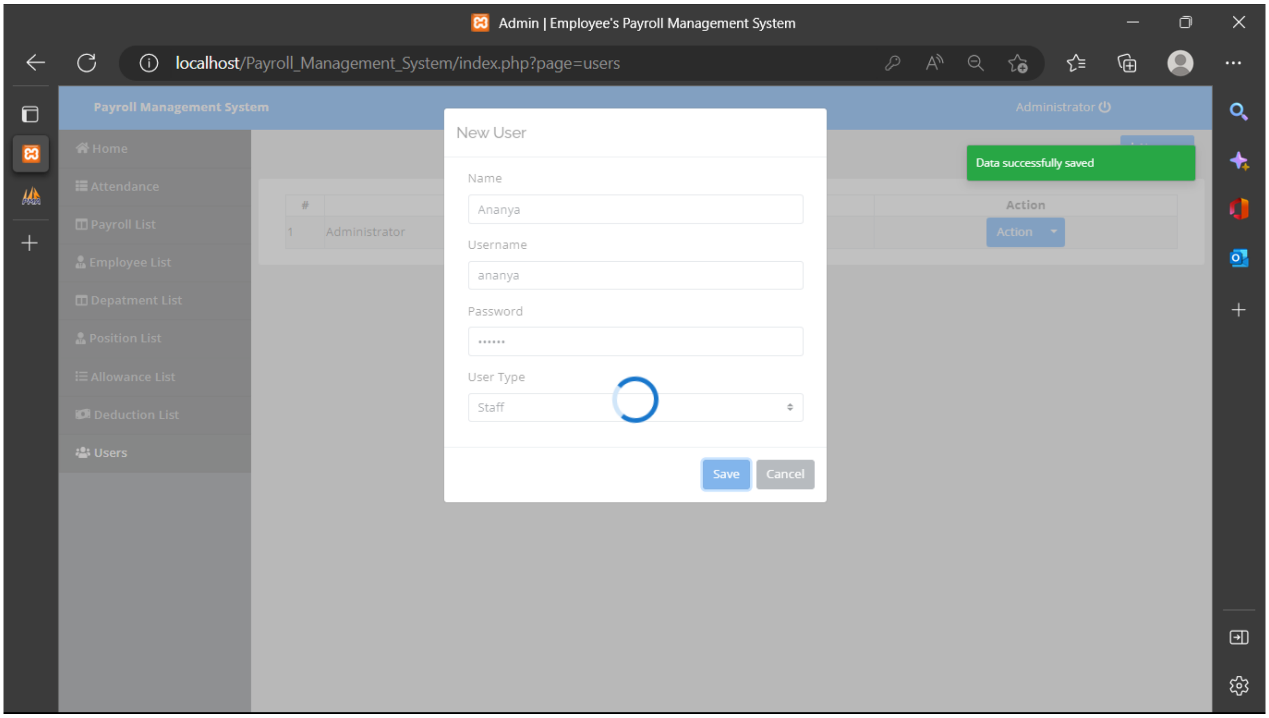
Results

	Trigger	Event	Table	Statement
0	before_employee_update	UPDATE	employee	INSERT INTO employees_audit SET action = 'update', emp
1	before_update	UPDATE	employee	INSERT INTO employees_audit SET action = 'update', emp
2	employee_update	UPDATE	employee	INSERT INTO employees_audit SET action = 'update', emp

Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table



localhost/Payroll_Management_System/index.php?page=users

Data successfully saved

New User

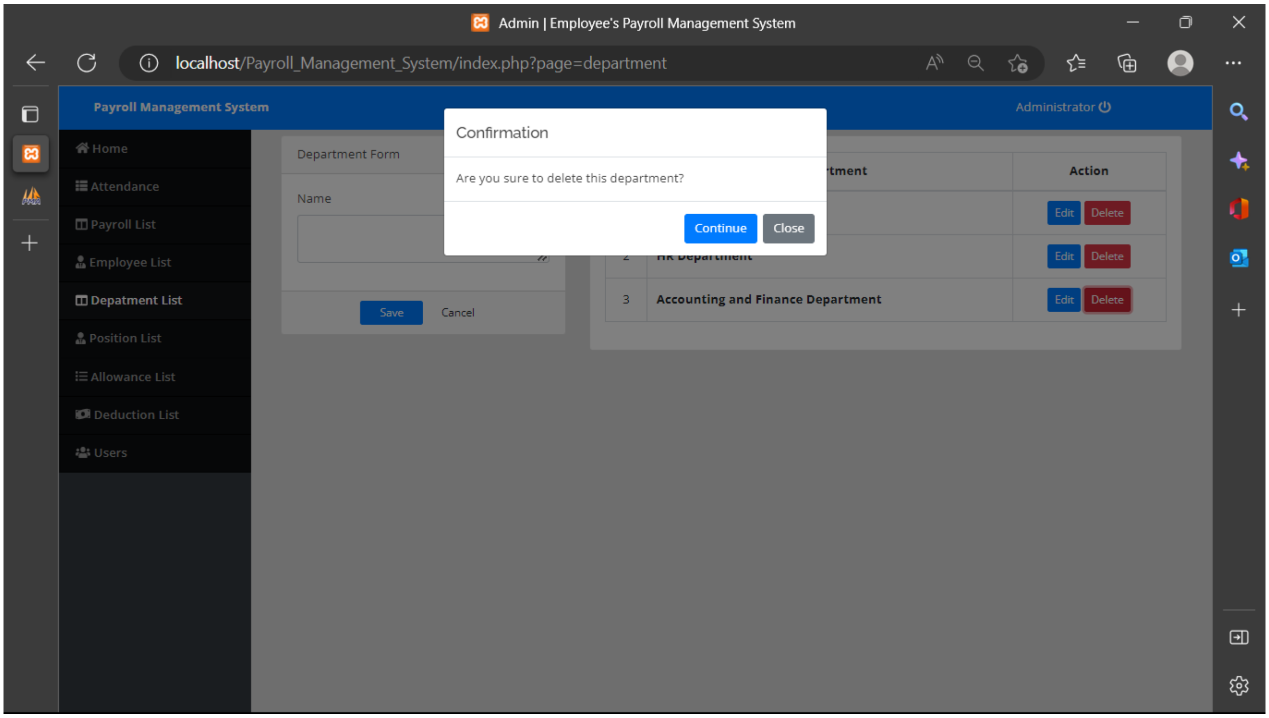
Name: Ananya

Username: ananya

Password: ••••••

User Type: Staff

#		Action
1	Administrator	Action



localhost/Payroll_Management_System/index.php?page=department

Confirmation

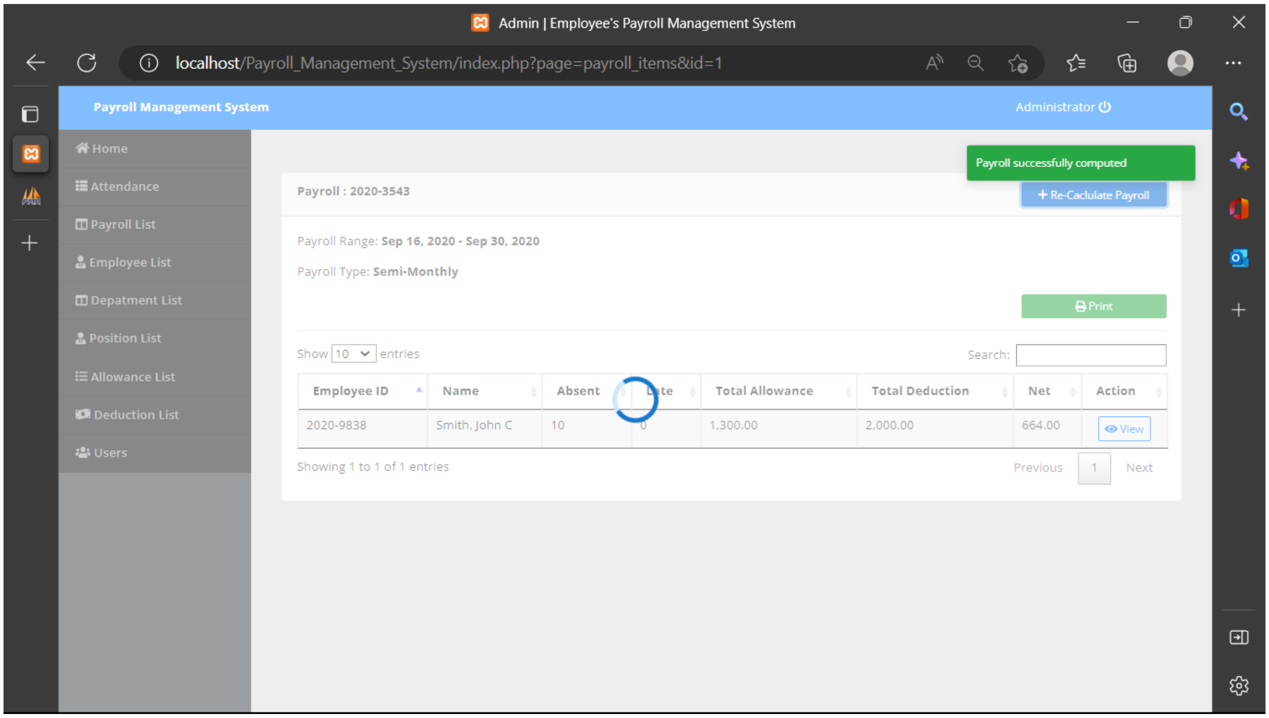
Are you sure to delete this department?

Department Form

Name

#	Department	Action
		Edit Delete
2	HR Department	Edit Delete
3	Accounting and Finance Department	Edit Delete

2. There should be an window to accept and run any SQL statement and display the result


Admin | Employee's Payroll Management System
localhost/Payroll_Management_System/index.php?page=payroll_items&id=1
Payroll Management System
Administrator
Home
Attendance
Payroll List
Employee List
Depatment List
Position List
Allowance List
Deduction List
Users
Re-Caclulate Payroll
Print

Payroll successfully computed

Payroll : 2020-3543

Payroll Range: Sep 16, 2020 - Sep 30, 2020

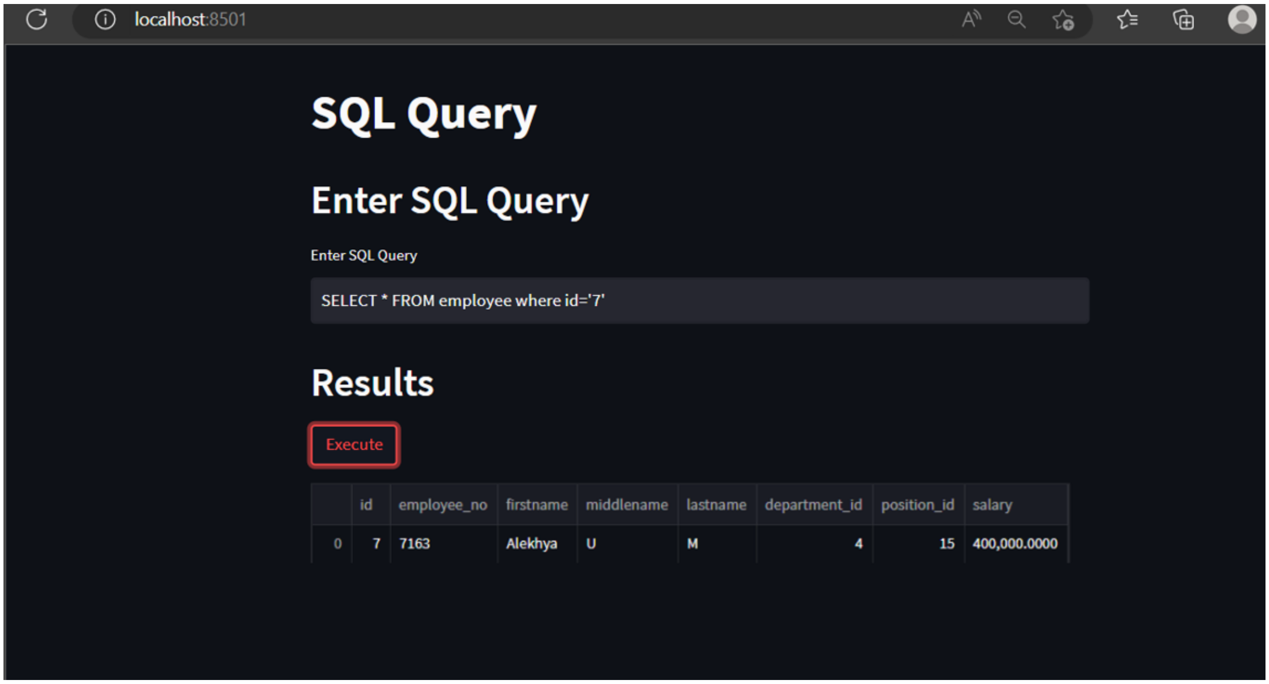
Payroll Type: Semi-Monthly

Show 10 entries Search:

Employee ID	Name	Absent	Late	Total Allowance	Total Deduction	Net	Action
2020-9838	Smith, John C	10	0	1,300.00	2,000.00	664.00	View

Showing 1 to 1 of 1 entries Previous 1 Next

QUERY BOX



SQL Query

Enter SQL Query

Enter SQL Query

```
SELECT * FROM employee where id='7'
```

Results

	id	employee_no	firstname	middlename	lastname	department_id	position_id	salary
0	7	7163	Alekhya	U	M	4	15	400,000.0000