Payroll Management System

Name: Utsav Bhatta
SFSU IF: 917597933
Git-hub ID: utsavb7420

Milestone/Version	Date
M1/V2	06/28/2020
M2/V1	07/16/2020

Table of Contents

Section I:

Project Description

This project ‘Payroll Management System’ aims at eliminating the manual intervention required for maintaining and calculating an employee’s salary based on their performance. This would enhance the performance of the HR department, but fulfill their requirements with the help of computerized equipment. By doing this, the data would be more secure and would be easily available across multiple platforms. The requirement software would be straightforward and easy to deal with. The Payroll Database Management system provides activities which are kept in the database for each employee record, so that all. The data is quicker and easier to retrieve. As the administrator 's standpoint, the administrator has the full right to make decisions like who all the users are and create, view, and modify statuses. This design and architecture is laid down in such a way as to eliminate and in some cases reduce the hardships faced by the existing system. This helps to administer the financial record of employees' salaries, wages, bonuses, net pay, increment, conveyance, loan and deductions. Moreover, this system is designed for needs of the company to carry out its operations in a smooth and effective manner.

Section II:

Use Cases

Use Case Title:	Works on a Project
Actors:	Employee
Description:	Hugh is an employee, an individual who can work on one more project assigned to him by his administrator which is available to his department. She has been hired by the company she is working on, and currently she is working on a field project within her software department. Some of his coworkers are working on the same project while some have been working on different projects.

Use Case Title:	Registers Attendance
Actors:	Employee
Description:	Katty was not feeling well today, due to the seasonal flu. She logs on the portal using her provided credentials and then she registers her attendance marking it as leave, knowing that this will impact her performance and hence salary. Her manager Hugh manages her attendance. She cannot take a leave unless her manager approves so. She learns that she is limited to one dependent.

Use Case Title:	Receives Salary
Actors:	Employee
Description:	It’s the end of the month, when Hugh logs into the portal to check on his salary, He is amazed that his payroll has been calculated accurately. He has the information on the basis of which this amount has been calculated. The detailed view of his bonus and leaves has been displayed.

Use Case Title:	Approves Leave
Actors:	Admin
Description:	Kahn signs on this portal and receives the notification that a few of his employees have requested for leaves today. Kahn took a sigh thinking that the rainy season has really outdone itself. He approves the leaves of his employees, and then sends them wish well cards their way.

Use Case Title:	Manages Salary
Actors:	Admin
Description:	Henry is the manager of his department, and it is his job to generate payroll slips. He logs on to the system to click the button which says “Generate Payslip”, and with that, everything is automatically calculated by the system.

Section III:

Database Requirements (Business Rules)

1. A department has one or more employees
2. An employee can have multiple specializations in different fields.
3. An employee can have multiple attendances, and that is managed by the employee's administrator.
4. An employee can work on zero or more projects, and these projects can be worked on by one more employee.
5. An employee can have one only one designation, but the same designation can be assigned to multiple employees.
6. A company has one or more departments.
7. Multiple departments can organize multiple events for the company. These can be workshops or training sessions.
8. A company can have multiple payrolls for which it is solely responsible to pay its employees.
9. An employee is managed by only one administrator at a given time.
10. An employee can receive multiple payroll receipts, and each of these belong to one employee only.
11. A department has different pay grades.
12. A payroll can have multiple deductions and bonuses.
13. A bonus is assigned to payroll on the basis of project.
14. Each project has an incentive in the form of a bonus, which belongs to that project only.
15. Admin is responsible for managing attendance for one or more employees.
16. Salary can have multiple payrolls which the employee then receives.

Section IV:

Detailed List of Main Entities, Attributes and Keys

1. **Employee** (Strong)
    - Emp_id; key, numeric
    - Name; candidate, alphanumeric
    - Dob; multivalue, timestamp
    - State; candidate, alphanumeric
    - City; candidate, alphanumeric
    - Email; candidate, alphanumeric
    - D_id ; key, numeric
    - Desg_id; key, numeric

2. **Admin** (Strong)
    - Admin_id, key, numeric
    - permission, alphanumeric
    - notes; alphanumeric
    - date_added; date

3. **Company** (Strong)
    - C_id; key, numeric
    - Name; candidate, alphanumeric
    - address; candidate, alphanumeric

4. **Department** (Strong)
    - D_id; key, numeric
    - C_id; key, numeric
    - Name; candidate, alphanumeric

5. **Specialization** (Strong)
    - C_id; key, numeric
    - Name; candidate, alphanumeric
    - type; alphanumeric
    - description; alphanumeric

6. **Event** (strong)
    - e_id; key, numeric
    - Name; candidate, alphanumeric
    - Date; multi-value, timestamp
    - Type_id; key, numeric
    - Address; candidate, alphanumeric

7. **Designation** (strong)
    - e_id; key, numeric
    - Name; candidate, alphanumeric
    - type; alphanumeric
    - description; alphanumeric

8. **Salary** (strong)
    - s_id; key, numeric
    - emp_id; key, numeric
    - start_date; candidate, date
    - end_date; candidate, date
    - admin_id; key, numeric
    - date; candidate, date
    - Hours_worked; candidate, numeric
    - b_id; key, numeric

9. **Payroll** (strong)
    - p_id; key, numeric
    - c_id; key, numeric
    - base_year; candidate, alphanumeric
    - description; candidate, alphanumeric

10. **Attendance** (strong)
    - att_id; key, numeric
    - admin_id; key, numeric
    - checkout; candidate, date
    - checkin; candidate, date
    - type; candidate, alphanumeric
    - status; candidate, alphanumeric

11. **Bonus** (strong)
    - b_id; key, numeric
    - amount; candidate, numeric
    - note; candidate, alphanumeric

12. **Deduction** (strong)
    - d_id; key, numeric
    - amount; candidate, alphanumeric
    - type; candidate, alphanumeric

13. **Paygrade** (Strong)
    - p_id; key, numeric
    - Name; candidate, alphanumeric
    - basic; candidate, alphanumeric
    - bonus; candidate, numeric

14. **Project** (Strong)
    - p_id; key, numeric
    - Name; candidate, alphanumeric
    - d_id; key, numeric
    - Type_id; key, numeric

15. **Attendance** (strong)
    - att_id; key, numeric
    - admin_id; key, numeric
    - checkout; candidate, date
    - checkin; candidate, date
    - type; candidate, alphanumeric
    - status; candidate, alphanumeric

16. **Event Type (Strong)**
    - type_id; key, number
    - location; key, alphanumeric
    - Description; alphanumeric

17. **Project Type (Strong)**
    - type_id; key, number
    - Description; alphanumeric
    - Details ; alphanumeric
    - Field; alphanumeric

18. **Bank (Strong)**
    - b_id; key, number
    - name; alphanumeric
    - address; alphanumeric

19. **Employee History (Strong)**
    - h_id; key, number
    - e_id; key, number
    - start_date; date
    - Experience; int

20. **Salary History (Strong)**
    - h_id; key, number
    - e_id; key, number
    - date; date
    - Salary; numeric

21. **Executive** (Strong)
    - Admin_id; key, numeric
    - Name; alphanumeric
    - Notes; text
    - Title; alphanumeric

22. **VP** (Strong)
    - v_id; key, numeric
    - Name; alphanumeric
    - Notes; text
    - Title; alphanumeric

23. **Payroll Department** (Strong)
    - p_id; key, numeric
    - Name; alphanumeric
    - Type: alphanumeric
    - Description; alphanumeric

24. **Insurance** (Strong)
    - i_id; key, numeric
    - Name; alphanumeric
    - Period;Date
    - registered_on;Date
    - Description; alphanumeric

25. **Employee_has_specializtion** (weak)
    - s_id; key, numeric
    - emp_id; key, numeric

26. **department_has_paygrades** (weak)
    - d_id; key, numeric
    - p_id; key, numeric

27. **department_has_events** (weak)
    - d_id; key, numeric
    - e_id; key, numeric

28. **project_has_bonus** (weak)
    - p_id; key, numeric
    - b_id; key, numeric

29. **payroll_has_bonus** (weak)
    - p_id; key, numeric
    - b_id; key, numeric

30. **payroll_has_deduction** (weak)
    - p_id; key, numeric
    - d_id; key, numeric

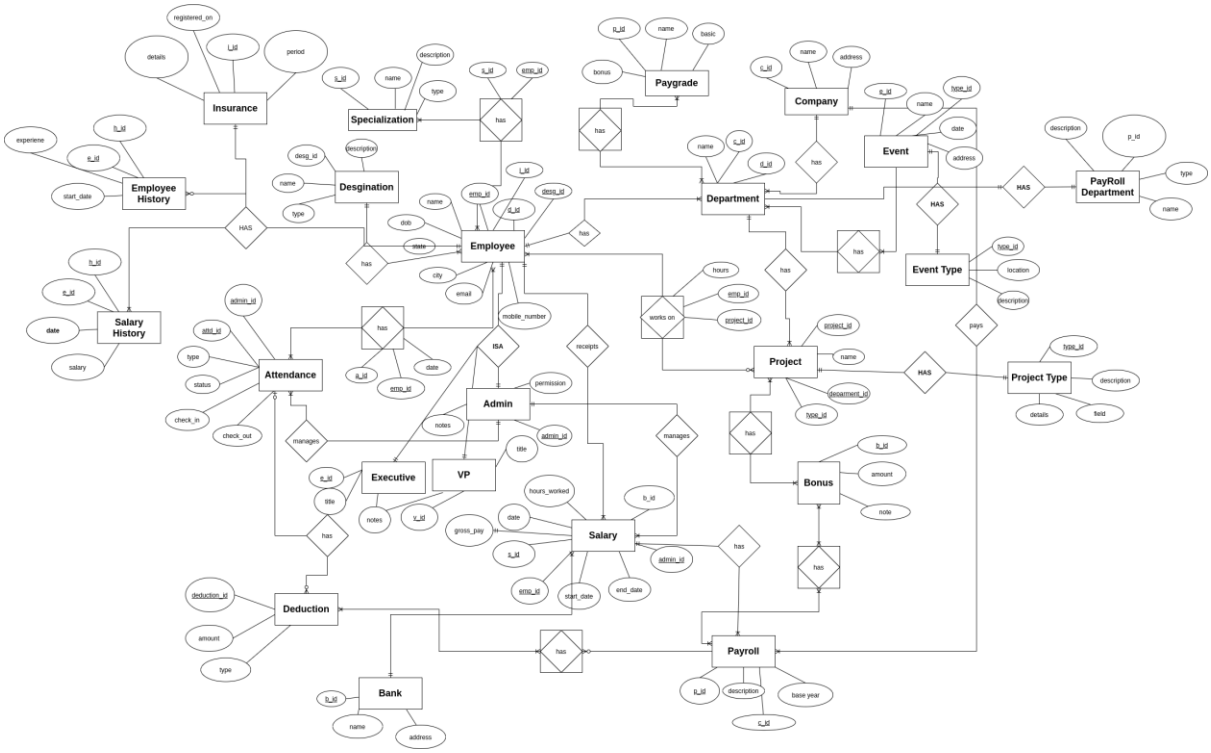
31. **employee_has_attendence** (weak)
    - e_id; key, numeric
    - a_id; key, numeric

32. **employee_has_project** (weak)
    - e_id; key, numeric
    - p_id; key, numeric

33. **Employee_has_specializtion** (weak)
    - s_id; key, numeric
    - emp_id; key, numeric

Section V:

Entity Relationship Diagram (ERD)



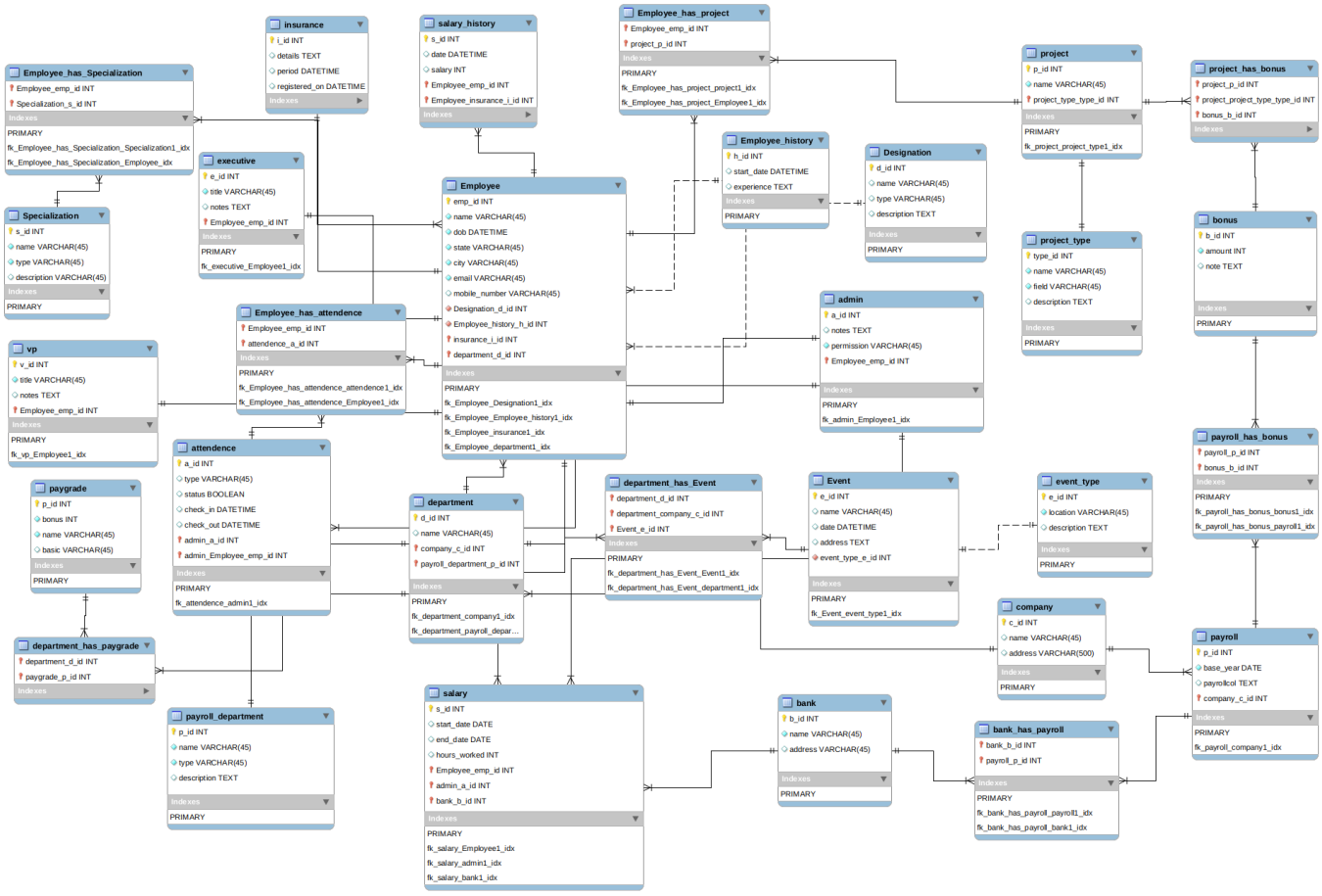
Section VI:

Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	Department	has	Employee	1 -M	Pass	N/A
2	Employee	specialises	Specialization	M-n	Pass	N/A
3	Employee	marks	Attendance	M-N	Pass	N/A
4	Employee	Works on	Project	0-1	Pass	N/A
5	Employee	Has	Designation	1-1	Pass	N/A
6	Company	Has	Department	1-M	Pass	N/A
7	Department	organises	Events	M-N	Pass	N/A
8	Company	has	Payroll	1-M	Fail	These payrolls should be unique for a company.
9	Admin	manages	Employee	1-M	Pass	N/A
10	Employee	receives	Payroll	1-M	Pass	N/A
11	Department	has	Paygrade	1-M	Pass	N/A
12	Payroll	has	Deduction	M-N	Pass	N/A
13	Payroll	Has	Bonus	M-N	Pass	N/A
14	Project	Has	Bonus	M-N	Pass	N/a
15	Salary	Has	Payroll	1-M	Pass	N/A
16	Admin	approves	Attendance	1-M	Pass	N/A

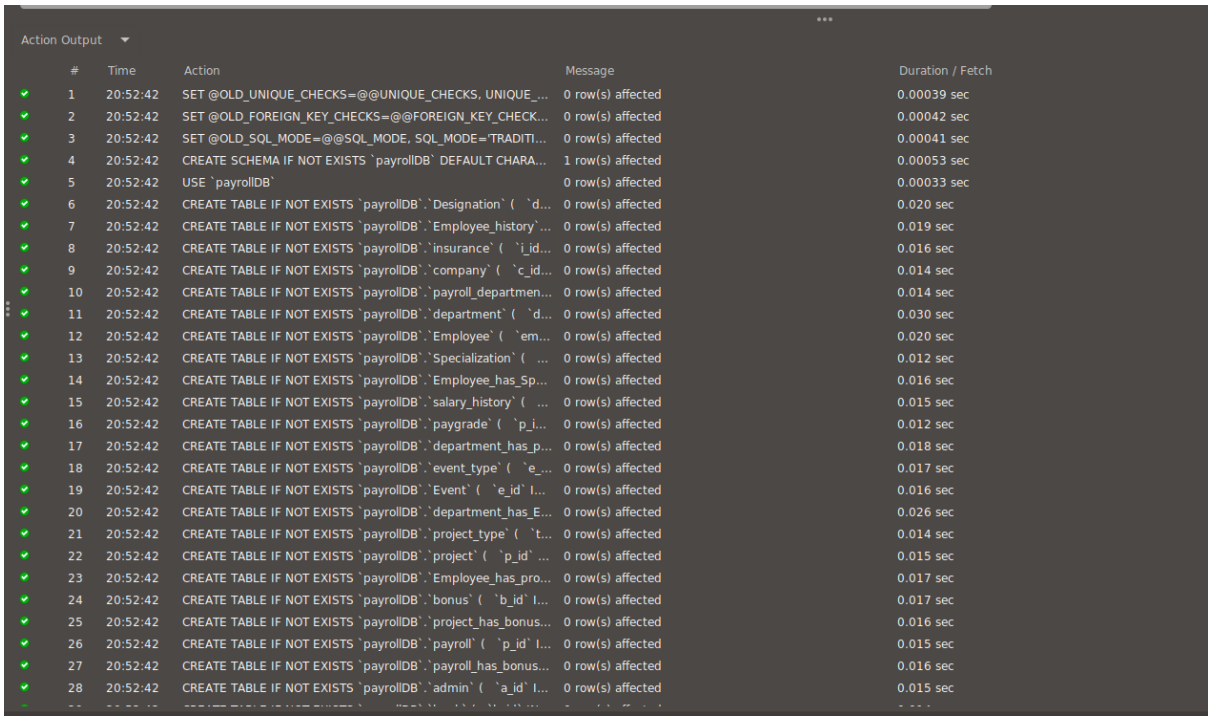
Section VII:

Database Model



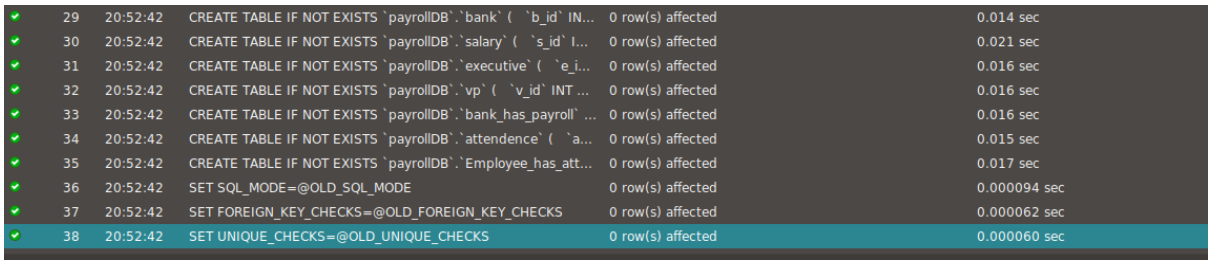
Section VIII:

Forward Engineering

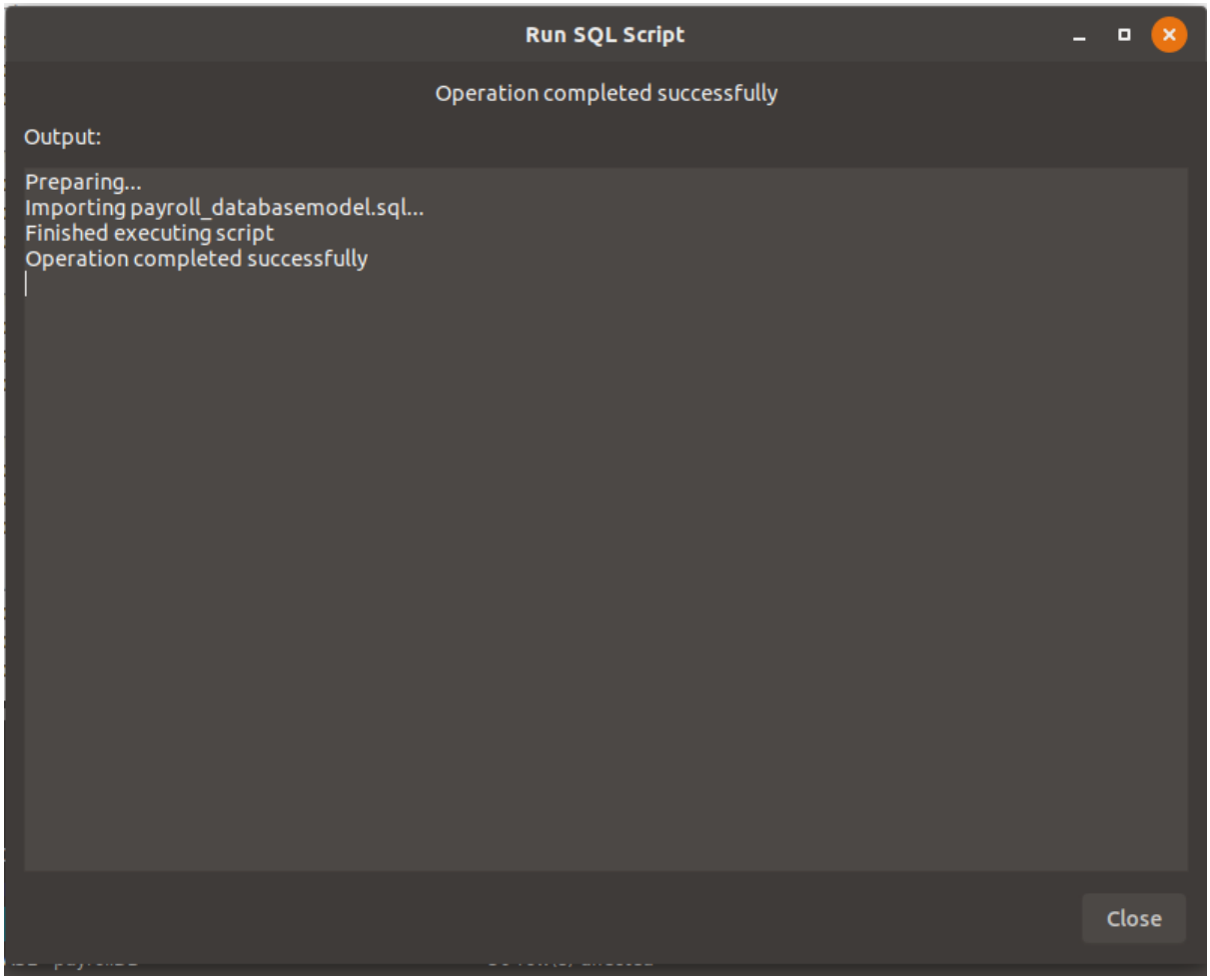


Action Output

#	Time	Action	Message	Duration / Fetch
1	20:52:42	SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_...	0 row(s) affected	0.00039 sec
2	20:52:42	SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECK...	0 row(s) affected	0.00042 sec
3	20:52:42	SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITI...	0 row(s) affected	0.00041 sec
4	20:52:42	CREATE SCHEMA IF NOT EXISTS `payrollDB` DEFAULT CHARA...	1 row(s) affected	0.00053 sec
5	20:52:42	USE `payrollDB`	0 row(s) affected	0.00033 sec
6	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Designation` (`d...	0 row(s) affected	0.020 sec
7	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Employee_history`...	0 row(s) affected	0.019 sec
8	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`insurance` (`i_id...	0 row(s) affected	0.016 sec
9	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`company` (`c_id...	0 row(s) affected	0.014 sec
10	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`payroll_departmen...	0 row(s) affected	0.014 sec
11	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`department` (`d...	0 row(s) affected	0.030 sec
12	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Employee` (`em...	0 row(s) affected	0.020 sec
13	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Specialization` (...	0 row(s) affected	0.012 sec
14	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Employee_has_Sp...	0 row(s) affected	0.016 sec
15	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`salary_history` (...	0 row(s) affected	0.015 sec
16	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`paygrade` (`p_i...	0 row(s) affected	0.012 sec
17	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`department_has_p...	0 row(s) affected	0.018 sec
18	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`event_type` (`e_...	0 row(s) affected	0.017 sec
19	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Event` (`e_id` I...	0 row(s) affected	0.016 sec
20	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`department_has_E...	0 row(s) affected	0.026 sec
21	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`project_type` (`t...	0 row(s) affected	0.014 sec
22	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`project` (`p_id` ...	0 row(s) affected	0.015 sec
23	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Employee_has_pro...	0 row(s) affected	0.017 sec
24	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`bonus` (`b_id` I...	0 row(s) affected	0.017 sec
25	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`project_has_bonus...	0 row(s) affected	0.016 sec
26	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`payroll` (`p_id` I...	0 row(s) affected	0.015 sec
27	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`payroll_has_bonus...	0 row(s) affected	0.016 sec
28	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`admin` (`a_id` I...	0 row(s) affected	0.015 sec

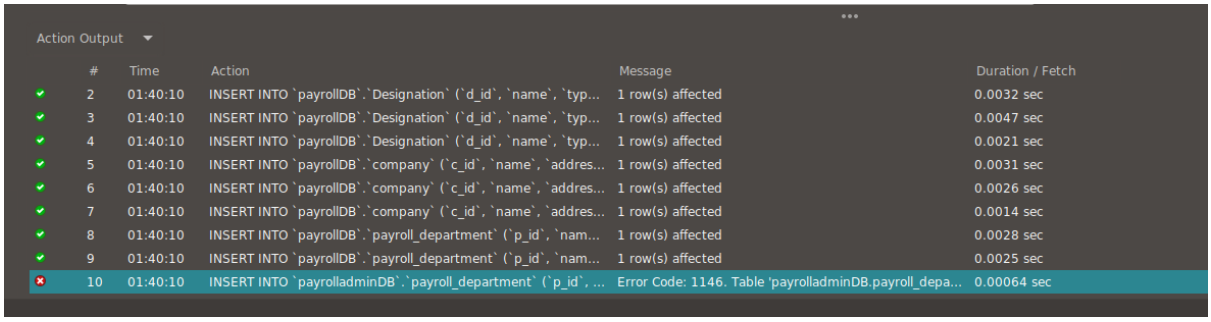


29	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`bank` (`b_id` IN...	0 row(s) affected	0.014 sec
30	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`salary` (`s_id` I...	0 row(s) affected	0.021 sec
31	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`executive` (`e_i...	0 row(s) affected	0.016 sec
32	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`vp` (`v_id` INT ...	0 row(s) affected	0.016 sec
33	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`bank_has_payroll` ...	0 row(s) affected	0.016 sec
34	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`attendence` (`a...	0 row(s) affected	0.015 sec
35	20:52:42	CREATE TABLE IF NOT EXISTS `payrollDB`.`Employee_has_att...	0 row(s) affected	0.017 sec
36	20:52:42	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected	0.000094 sec
37	20:52:42	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected	0.000062 sec
38	20:52:42	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	0.000060 sec


Run SQL Script
Operation completed successfully
Output:
Preparing...
Importing payroll_databasemodel.sql...
Finished executing script
Operation completed successfully
Close

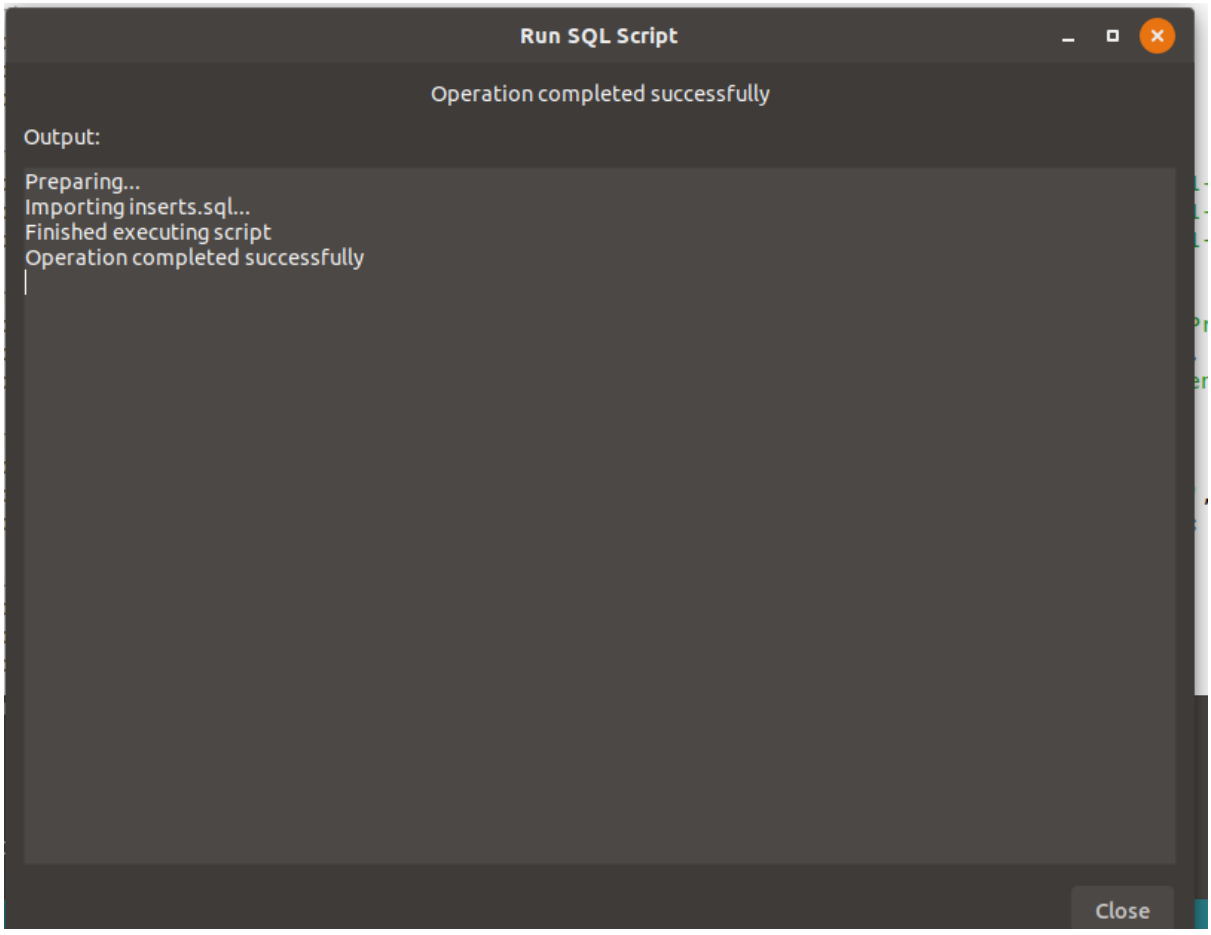
Section IX:

Inserting Data



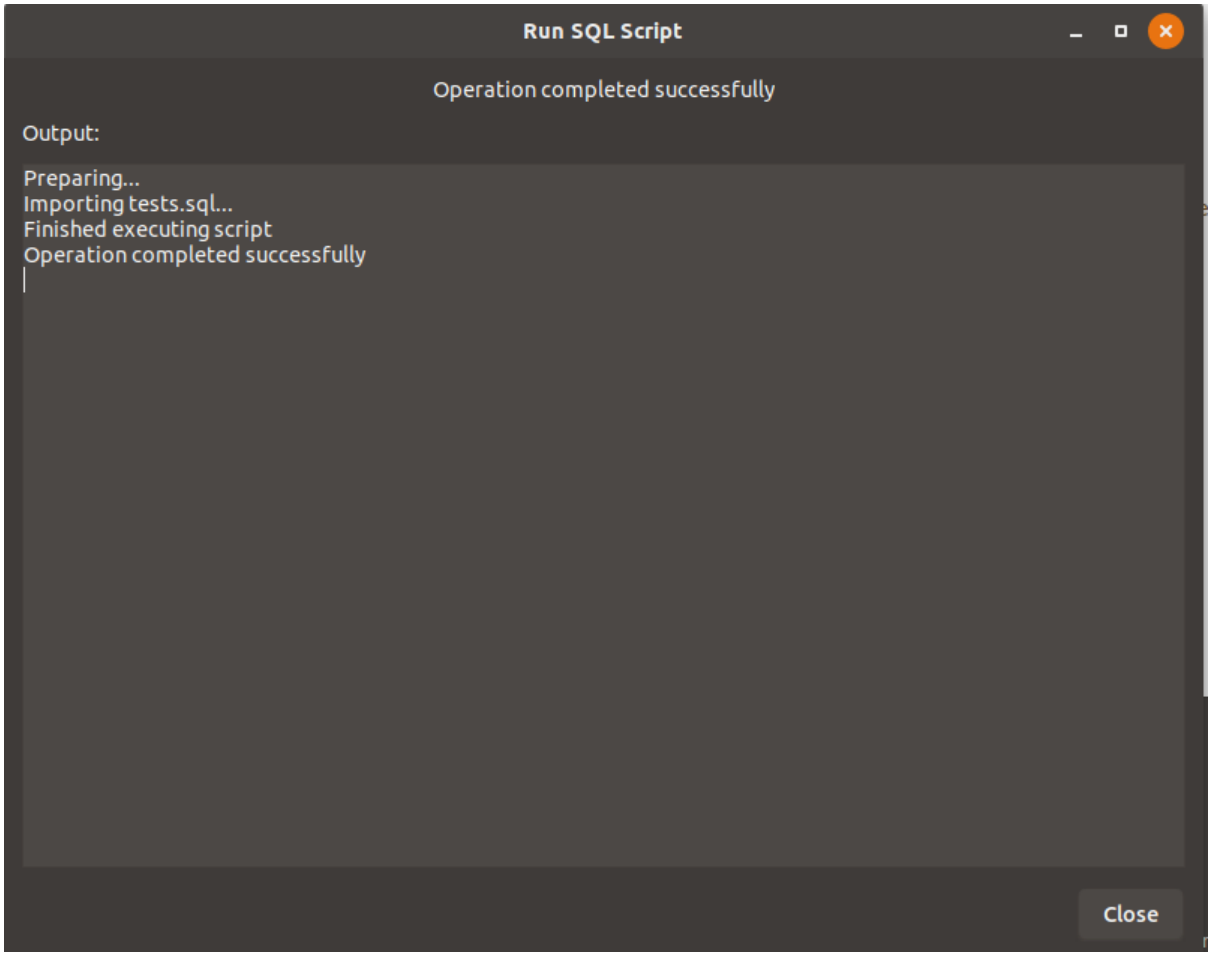
Action Output

#	Time	Action	Message	Duration / Fetch
2	01:40:10	INSERT INTO `payrollDB`.`Designation` (`d_id`, `name`, `typ...	1 row(s) affected	0.0032 sec
3	01:40:10	INSERT INTO `payrollDB`.`Designation` (`d_id`, `name`, `typ...	1 row(s) affected	0.0047 sec
4	01:40:10	INSERT INTO `payrollDB`.`Designation` (`d_id`, `name`, `typ...	1 row(s) affected	0.0021 sec
5	01:40:10	INSERT INTO `payrollDB`.`company` (`c_id`, `name`, `addres...	1 row(s) affected	0.0031 sec
6	01:40:10	INSERT INTO `payrollDB`.`company` (`c_id`, `name`, `addres...	1 row(s) affected	0.0026 sec
7	01:40:10	INSERT INTO `payrollDB`.`company` (`c_id`, `name`, `addres...	1 row(s) affected	0.0014 sec
8	01:40:10	INSERT INTO `payrollDB`.`payroll_department` (`p_id`, `nam...	1 row(s) affected	0.0028 sec
9	01:40:10	INSERT INTO `payrollDB`.`payroll_department` (`p_id`, `nam...	1 row(s) affected	0.0025 sec
10	01:40:10	INSERT INTO `payrolladminDB`.`payroll_department` (`p_id`, ...	Error Code: 1146. Table 'payrolladminDB.payroll_depa...	0.00064 sec



Section X:

Testing



Section XI:

Testing Table

Entity	SQLQuery	OK/Failed	Error Description	Possible Solution
Designation	DELETE	OK	None	None
Designation	UPDATE	OK	None	None
Company	DELETE	OK	None	None
Company	UPDATE	OK	None	None
Payroll_department	DELETE	OK	None	None
Payroll_department	UPDATE	OK	None	None
Department	DELETE	OK	None	None
Department	UPDATE	OK	None	None
Bank	DELETE	OK	None	None

Bank	UPDATE	OK	None	None
Payroll	DELETE	OK	None	None
Payroll	UPDATE	OK	None	None
project_type	DELETE	OK	None	None
project_type	UPDATE	OK	None	None
project	DELETE	FAILED	Value not found	forgot to truncate the tables, and insert again
project	UPDATE	FAILED	Value not found	forgot to truncate the tables, and insert again
bank_has_payroll	DELETE	OK	None	None
bank_has_payroll	UPDATE	OK	None	None
bonus	UPDATE	OK	None	None
bonus	DELETE	FAILED	Cannot delete or update a parent row: a foreign key constraint fails (`payrollDB`.`payroll_has_bonus`, CONSTRAINT `fk_payroll_has_	forgot to truncate the tables, and insert again

			bonus_bonus1`	
Specialization	DELETE	FAILED	: Cannot delete or update a parent row: a foreign key constraint fails (`payrollDB`.`Employee_has_Specialization`	forgot to truncate the tables, and insert again
Specialization	UPDATE	OK	None	None
Insurance	DELETE	Ok	NONE	None
Insurance	UPDATE	OK	None	None
employee_history	DELETE	Ok	None	None
employee_history	UPDATE	Ok	None	None
employee	DELETE	Ok	None	None
employee	UPDATE	Ok	None	None
attendance	DELETE	Ok	None	None
attendance	UPDATE	Ok	None	None

event_type	DELETE	Ok	None	None
event_type	UPDATE	Ok	None	None
event	DELETE	Ok	None	None
event	UPDATE	Ok	None	None
departmnet_has_event	DELETE	Ok	None	None
departmnet_has_event	UPDATE	Ok	None	None
employee_has_attendence	DELETE	Ok	None	None
employee_has_attendence	UPDATE	Ok	None	None