

Employee Payroll and Performance Database

Schema Overview/ Design

Abdul Basit [200757150]
Lubna Al Rifaie [200821590]
Harriharan Sivakumar[200676770]
Lashani Dharmasena [212378870]

Table of Contents

- Project Overview
 - Problem and Solution (project goals)
- DB Schema Overview
 - Entities
 - Relationships
 - Attributes
- Implementation
 - Workbench:
 - Tables
 - Constraints and Domain Constraints
 - SQL Commands (mainly create)

Project Overview

The focus of this project is to design a database that is used to manage employee performance and payroll.

The two main subcomponents:

1. Employee payroll administration
2. Employee Performance Management

The users can interact using a web application for employees to manage

Entities and relationships

Overview of Entities and the relationship among them

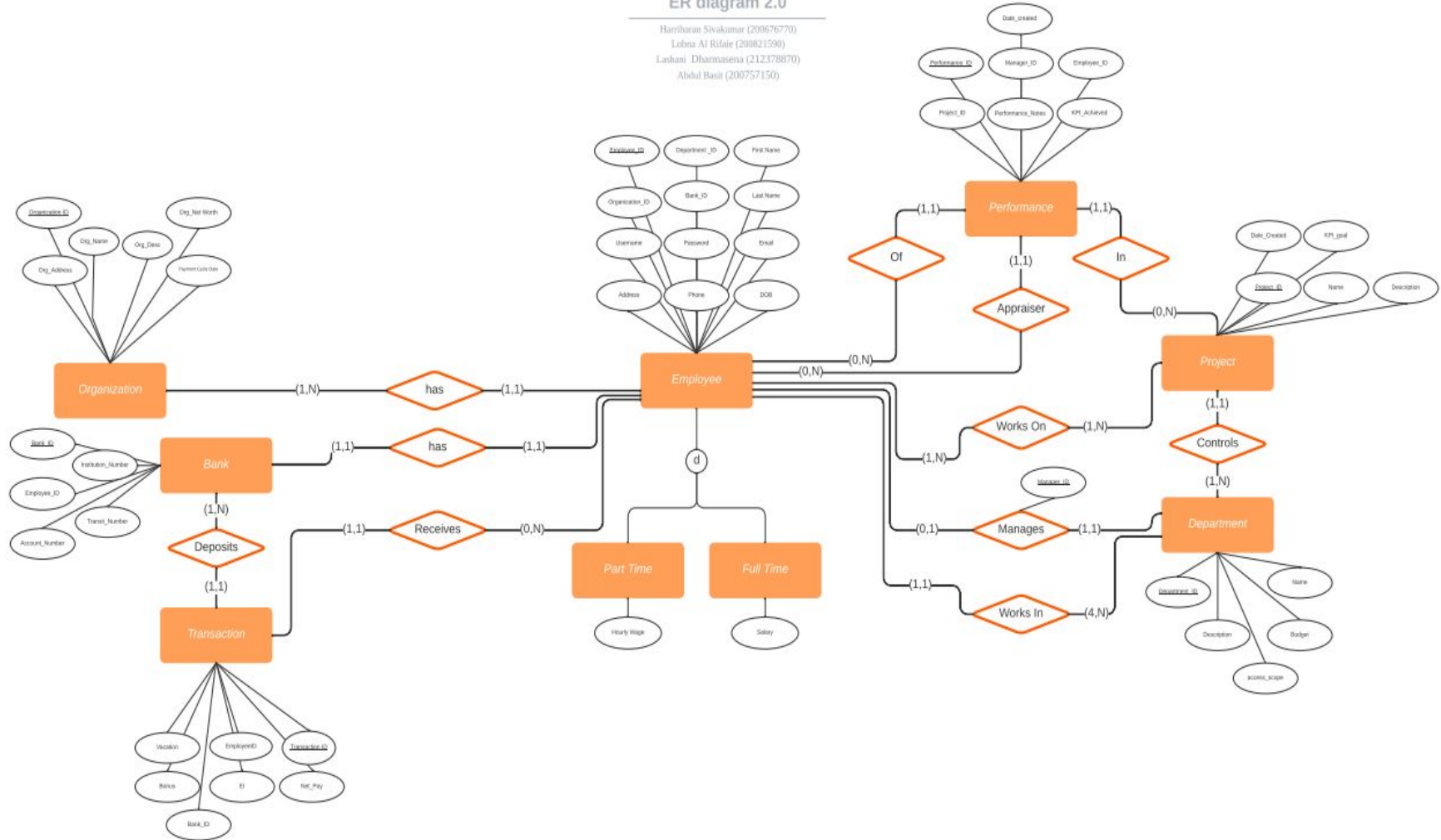
Entities:

- Organization
- Employee
- Performance
- Project
- Department
- Bank
- Transaction

The ER Diagram

ER diagram 2.0

Hariluan Sivakumar (200676770)
 Leha Al Rifale (200821590)
 Ladhani Dharmasena (212378870)
 Abdul Basit (200757150)

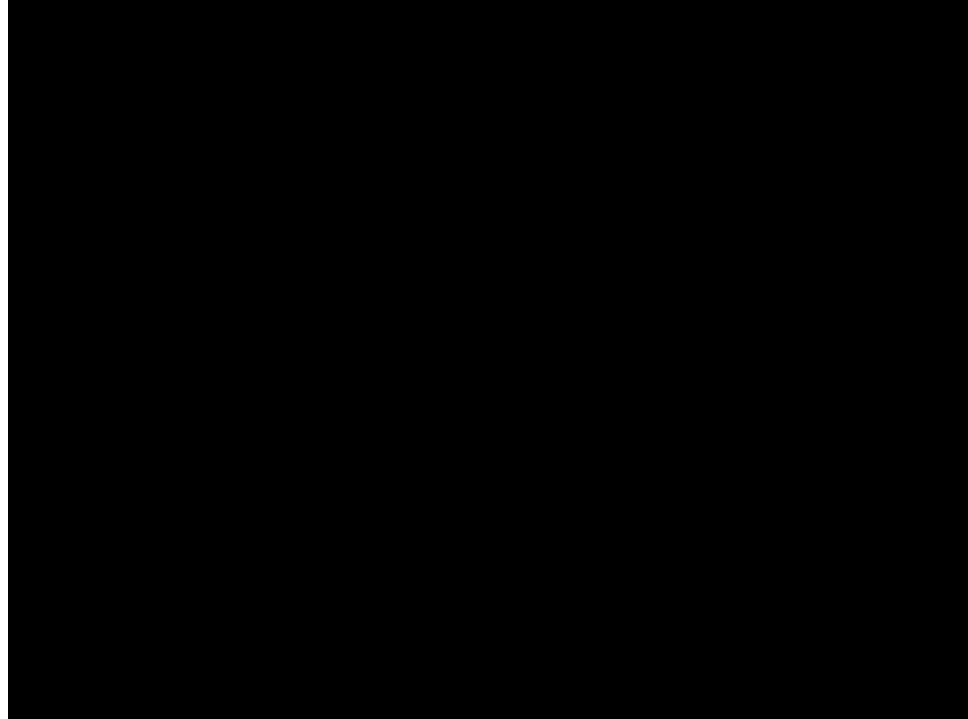


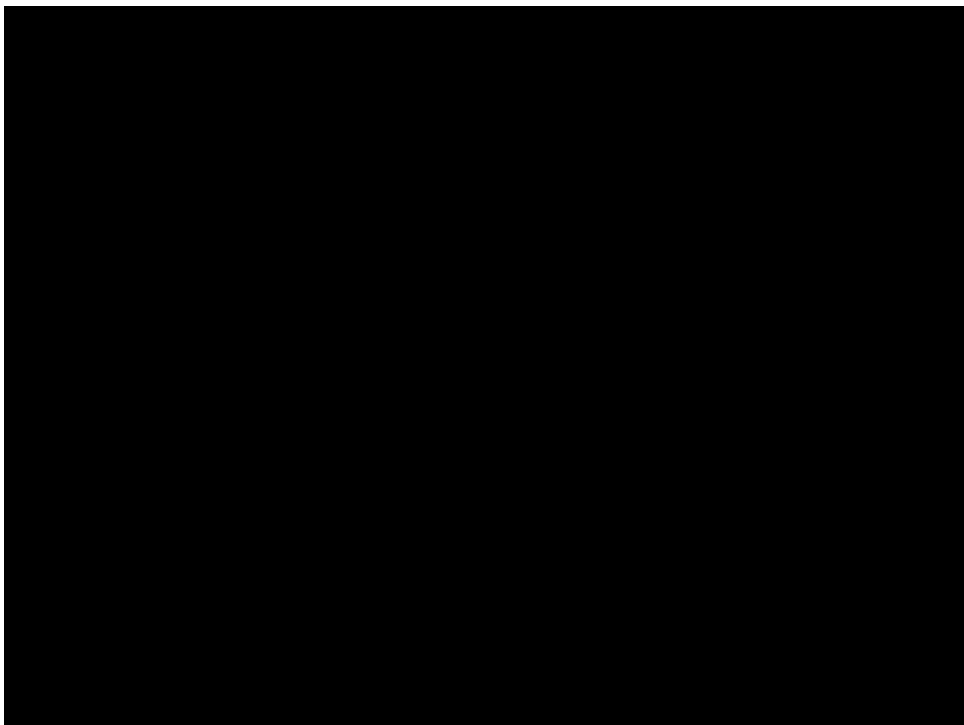
SQL Commands

How we created the tables

Create, Select, Insert Commands
(show results in video as well)

Show that the domain constraints are
implemented correctly with the help of
workbench SS






```

11
12 • CREATE TABLE Employee(
13     employee_id INT PRIMARY KEY AUTO_INCREMENT,
14     emp_firstName VARCHAR(50) NOT NULL,
15     emp_lastName VARCHAR(50) NOT NULL,
16     emp_address TEXT NOT NULL,
17     emp_phone VARCHAR(15) NOT NULL,
18     emp_username VARCHAR(50) NOT NULL UNIQUE,
19     emp_email VARCHAR(50) NOT NULL UNIQUE,
20     emp_password VARCHAR(50) NOT NULL,
21
22     -- Disjoint constraint
23     emp_type ENUM('full_time', 'part_time') NOT NULL, #employee can either be part time or full time
24     emp_hourlyWage DECIMAL(10,2) DEFAULT NULL,
25     emp_salary DECIMAL(10,2) DEFAULT NULL,
26
27     CHECK(
28         # if FULL-TIME then salary is NOT null and hourly rate is null
29         (emp_type = 'full_time' AND emp_salary IS NOT NULL AND emp_hourlyWage IS NULL)
30         OR
31         # if PART-TIME then salary is null and hourly rate is NOT null
32         (emp_type = 'part_time' AND emp_salary IS NULL AND emp_hourlyWage IS NOT NULL)
33     ),
34
35     -- Foreign Keys
36     organization_id INT NOT NULL,
37     department_id INT NOT NULL,
38     bank_id INT NOT NULL,
39     FOREIGN KEY (organization_id) REFERENCES Organizations(organization_id),
40     FOREIGN KEY (department_id) REFERENCES Department(department_id),
41     FOREIGN KEY (bank_id) REFERENCES Bank(bank_id)
42 );

```

```
1 • create database cp363finaldatabase;
2
3 • CREATE TABLE Organizations(
4     organization_id INT PRIMARY KEY AUTO_INCREMENT, #primary key that will auto increment
5     org_name VARCHAR(80) NOT NULL UNIQUE,           #organization name w/ constraints of being unique and not null
6     org_address TEXT,
7     org_desc TEXT,
8     org_netWorth DECIMAL(10,2),
9     payment_cycle DATE NOT NULL
10 );
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44 • CREATE TABLE Bank(
45     bank_id INT PRIMARY KEY AUTO_INCREMENT,
46     institute_number VARCHAR(50) NOT NULL UNIQUE,
47     transit_number VARCHAR(50) NOT NULL UNIQUE,
48     account_number VARCHAR(50) NOT NULL UNIQUE,
49
50     -- Foreign Keys
51     employee_id INT NOT NULL,
52     FOREIGN KEY (employee_id) REFERENCES Employee(employee_id)
53 );
54
```

```

54
55 • CREATE TABLE Transactions(
56     transaction_id INT PRIMARY KEY AUTO_INCREMENT,
57     transaction_date DATE NOT NULL,
58     bonus_pay DECIMAL(10,2) CHECK (bonus_pay >= 0),           #domain constraint making sure value >= 0
59     EI_pay DECIMAL(10,2) CHECK (EI_pay >= 0),
60     net_pay DECIMAL(10,2) NOT NULL CHECK (net_pay >= 0),
61     vacation_pay DECIMAL(10,2) CHECK (vacation_pay >= 0),
62
63     -- Foreign Keys
64     employee_id INT NOT NULL,
65     bank_id INT NOT NULL,
66     FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
67     FOREIGN KEY (bank_id) REFERENCES Bank(bank_id)
68 );
69
70 • CREATE TABLE Performance(
71     performance_id INT PRIMARY KEY AUTO_INCREMENT,
72     perf_notes TEXT NOT NULL,
73     date_Achieved DATE NOT NULL,
74     KPI_Achieved INT NOT NULL CHECK (KPI_Achieved >= 0 AND KPI_Achieved <= 100),
75
76     -- Foreign Keys
77     manager_id INT NOT NULL,
78     employee_id INT NOT NULL,
79     project_id INT NOT NULL,
80
81     FOREIGN KEY (manager_id) REFERENCES Employee(employee_id),
82     FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
83     FOREIGN KEY (project_id) REFERENCES Project(project_id)
84 );

```

```
85
86 • CREATE TABLE Project(
87     project_id INT PRIMARY KEY AUTO_INCREMENT,
88     prj_name VARCHAR(50) NOT NULL,
89     prj_desc TEXT NOT NULL,
90     date_created DATE NOT NULL,
91     KPI_goal INT NOT NULL CHECK (KPI_goal >= 0 AND KPI_goal <= 100) # makes sure our KPI is a "percentage" out of 100%
92 );
93
94 • CREATE TABLE Department(
95     department_id INT PRIMARY KEY AUTO_INCREMENT,
96     dept_name VARCHAR(50) NOT NULL UNIQUE,
97     dept_desc TEXT NOT NULL,
98     dept_budget DECIMAL(10,2) NOT NULL CHECK (dept_budget >= 0),
99
100     -- Foreign Keys
101     manager_id INT NOT NULL,
102     FOREIGN KEY (manager_id) REFERENCES Employee(employee_id) #since a manager is just an employee, we reference an employee here as a "manager" of a department
103
104 );
105
106
```
