

DBMS ASSIGNMENT

Roll No: 210710007037

Department of Computer Science and Engineering, JEC

Date of Submission: 21.08.2023

ASSIGNMENT 1:

Design a database schema with a brief on its purpose/functionality and 5-7 relations. Further, populate the relations with 5 tuples instance.

Soln:

Database Schema: Employee Management System

Purpose: The Employee Management System (EMS) database is designed to store and manage information about employee within an organization. It aims to streamline HR processes, track employee details, monitor attendance records, among various, thereby facilitating effective management of human resources.

This database includes 5 relations and their subsequent attributes, including primary key (PK) and foreign key (FK). The relations are further populated with 5 tuples instances each, which are randomly synthesized.

Relations (Attributes):

1. employees (employee_id, name, date_of_birth, gender, department_id, position_id)
2. positions (position_id, position_title, salary)
3. departments (department_id, department_name, location)
4. salaries (salary_id, employee_id, salary_amount, effective_data)
5. attendance (attendance_id, employee_id, date_of_attendance, time_in, time_out)

A schema diagram, also known as a database schema diagram or simply a schema, is a visual representation of the structure of a database system. It illustrates the logical organization of data within a database, including the tables, columns, relationships, constraints, and keys.

The primary keys are denoted as underlined attributes.

The foreign keys are denoted using arrows.

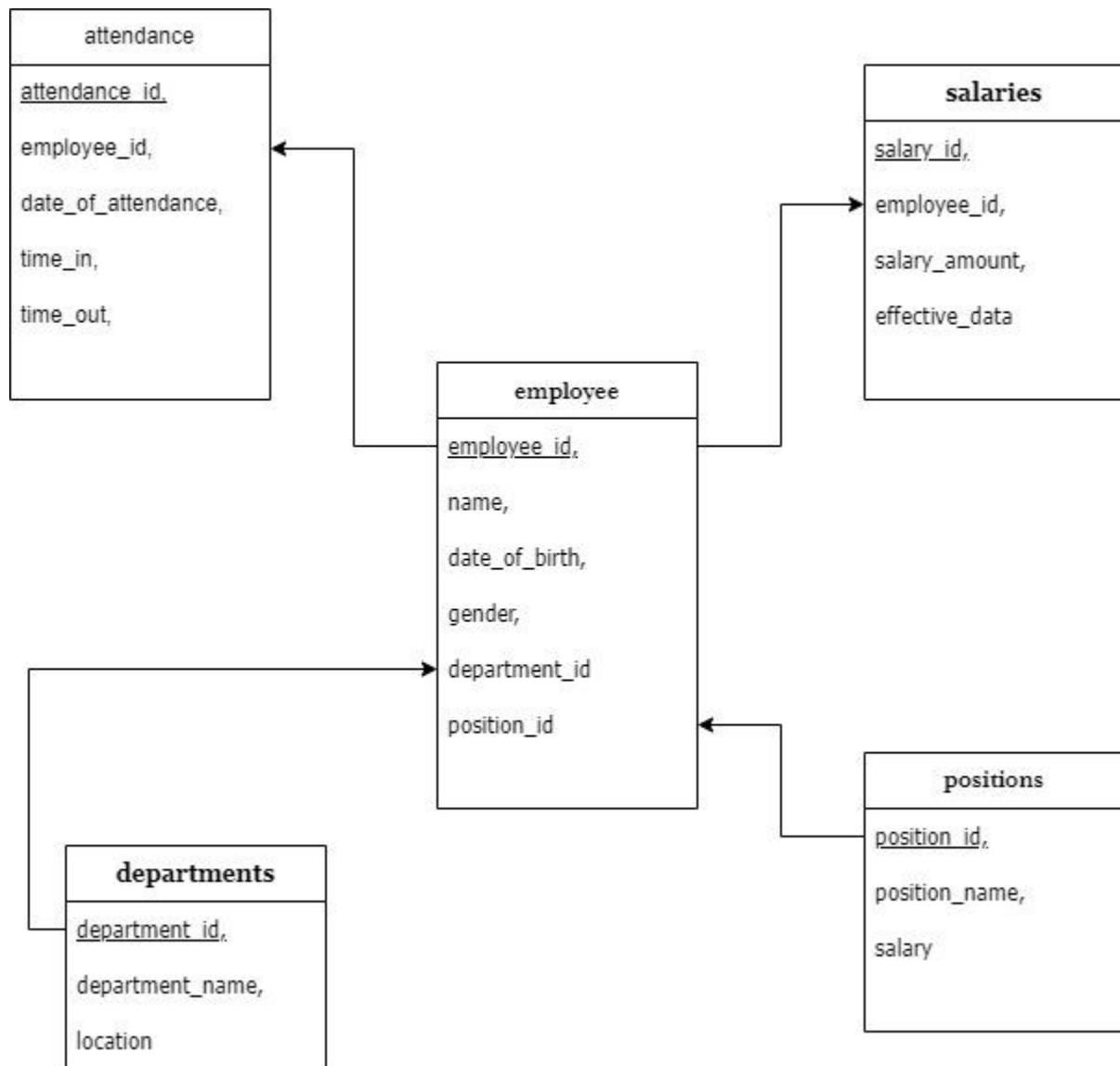


Fig: Schema Diagram of the EMS database

5 tuples instances for each relation:

Employee:

	employee_id	name	date_of_birth	gender	department_id	positions_id
▶	101	John Doe	03-15-1985	Male	301	201
	102	Jane Smith	20-09-1990	Female	302	202
	103	Michael Johnson	10-06-1988	Male	303	203
	104	Emily Brown	05-12-1992	Female	304	204
	105	William Wilson	02-08-1987	Female	305	205
★	NULL	NULL	NULL	NULL	NULL	NULL

Departments:

	department_id	department_name	location
▶	301	Human Resources	Building A
	302	Marketing	Building B
	303	Engineering	Building C
	304	Finance	Building A
	305	Sales	Building D
★	NULL	NULL	NULL

Positions:

	position_id	position_title	salary
▶	201	HR Manager	70000
	202	Marketing Specialist	55000
	203	Software Engineer	80000
	204	Financial Analysis	60000
	205	Sales Representative	50000
★	NULL	NULL	NULL

Salaries:

	salary_id	employee_id	salary_amount	effective_data
▶	401	101	70000	01-01-2023
	402	102	60000	01-01-2023
	403	103	80000	01-01-2023
	404	104	65000	01-01-2023
	405	105	55000	01-01-2023
★	NULL	NULL	NULL	NULL

Attendance:

	attendance_id	employee_id	date_of_attendance	time_in	time_out
▶	501	101	10-08-2023	09:00 AM	06:00 PM
	502	102	10-08-2023	09:15 AM	06:15 PM
	503	103	10-08-2023	09:30 AM	06:30 PM
	504	104	10-08-2023	09:00 AM	06:00 PM
	505	105	10-08-2023	09:15 AM	06:15 PM
✱	NULL	NULL	NULL	NULL	NULL

ASSIGNMENT 2:

Left outer join; Right outer join

Soln:

Left outer join: A left outer join is a type of join operation in relational databases that combines rows from two tables based on a common column (or columns), while also including all the rows from the left table, even if there is no match in the right table.

Let us consider two tables: "Customers" and "Orders".

The "Customers" table contains attributes like customer_id, name and email. The "Orders" table contains attributes like order_id, customer_id, and order_date. If we perform a left outer join between the "Customers" table (left table) and "Orders" table (right table) based on the customer_id attribute, the result will include all the rows from the "Customers" table, along with the matching rows from the "Orders" table. If there is no matching row in the "Orders" table for a particular customer, the result will still include the customer's information, but the columns related to the orders will contain NULL values.

Right outer join: A right outer join is similar to a left outer join, but it includes all rows from the right table and matching rows from the left table. All rows from the right table are included in the result, even if there are no matching rows in the left table.

Considering the same two tables' "Customers" and "Orders",

Performing a right outer join between the "Customers" table (left table) and the "Orders" table (right table) based on the customer_id attribute would result in including all rows from the "Orders" table and matching rows from the "Customers" table. If there is no matching row in the "Customers" table for a particular order, the result will still include that order's information, but the columns related to customers will contain NULL values.