



**LAU**  
**School of Arts and Sciences**

*Department of Computer Science & Mathematics*

**Lebanese American University**

**School of Arts and Sciences**

**Department of Computer Science and Mathematics**

**A Database Design for HR Management System**

**Report Done By “Data Crafters” Group**

**Mostapha Zeineddin**

[mostapha.zeineddin@lau.edu](mailto:mostapha.zeineddin@lau.edu)

**Jad Al-Awar**

[jad.alawar@lau.edu](mailto:jad.alawar@lau.edu)

**Ryan Toulujian**

[ryan.toulujian@lau.edu](mailto:ryan.toulujian@lau.edu)

Submitted to Dr. Abbas Bassam in partial fulfillment of the requirements of  
phase 4 of the database project for the CSC375 (Database Management Systems) course Spring 2024  
semester

April 2024

# Table of Contents

I.	Introduction.....	3
II.	Copyright Notice.....	4
III.	ER Legend.....	5
IV.	ER Diagram.....	6
V.	Entities and their attributes.....	7-16
1.	EMPLOYEE.....	7
2.	DEPARTMENT.....	8
3.	RECRUITMENT.....	9
4.	APPLICANT.....	10
5.	PAYROLL.....	11
6.	ATTENDANCE.....	12
7.	VACATION.....	13
8.	EVALUATION.....	14
9.	BENEFITS.....	15
10.	DEPENDENT.....	16
VI.	Relationships.....	17-23
1.	WORKS_FOR.....	17
2.	MANAGES.....	17
3.	PUBLISHES.....	18
4.	ASSESES.....	18
5.	APPLIES.....	19
6.	HAS.....	19
7.	UNDERGOES.....	20
8.	LOGS_FOR.....	20
9.	ELIGIBLE_FOR.....	21
10.	DOES.....	21
11.	DEPENDENTS_OF.....	22
12.	RESPONSIBLE_FOR.....	22
13.	HAS.....	23
VII.	Mapping.....	24-31
VIII.	Creation of Tables.....	32-39
IX.	Insertion of Data.....	39-55
X.	SQL Queries.....	55-64
XI.	Normalization.....	65-73
XII.	Conclusion.....	74
XIII.	Instructor's Feedback.....	75

## **I. Introduction**

After a massive storm hit, it caused a lot of damage. One of the biggest problems was that it completely destroyed the Human Resources (HR) stored database and servers of the MSD company. This was a huge issue because the HR database had all the important information about the company's employees, like their names, their departments, and contact details. Without this database, the company was in trouble. MSD knew it had to do something quick and perfect to fix this problem, so they decided to make a brand-new HR database, but this time, they wanted to make it even better than before. They came up with a plan to contact Data Crafters and order a database for HRMS that must be clear and understandable so it can be easier to keep track of everyone and their roles. To make this new system, we as Data Crafters, started by designing a new ER (Entity-Relationship) diagram. An ER diagram is like a map that shows how all the different parts of a database are connected. This new diagram was carefully made to make sure the new database would be strong and able to protect the employees' private information. With a lot of hard work and smart thinking, the Crafters were able to build the new HR database. It was not only just as good as the old one, but it was even better. It had new features that made it more secure and better at keeping employees' information safe. This new system was a huge step forward for MSD. It helped them recover quickly from the storm's damage and showed how important it is to have a good system for managing data.



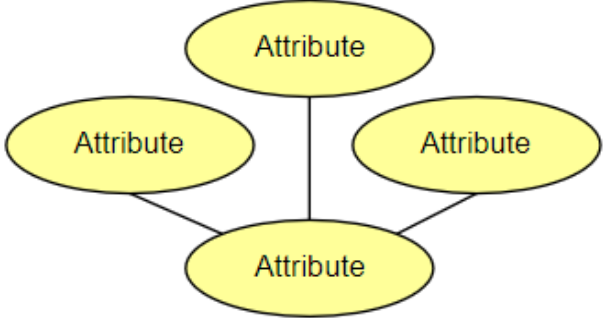


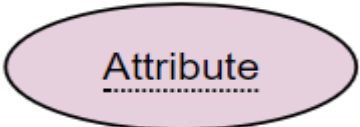

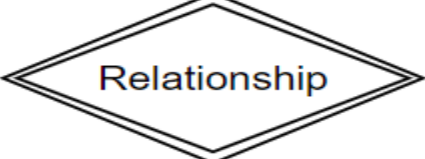
## **II. Copyright Notice**

Copyright © 2024 "Data Crafters". All rights reserved

Data Crafters, operating under the HR system: (CraftersHR: MSD HR System), this database project, including its design, architecture, and content, is protected by copyright laws and international treaties. Unauthorized reproduction or distribution of any part of this project may result in civil and criminal penalties.

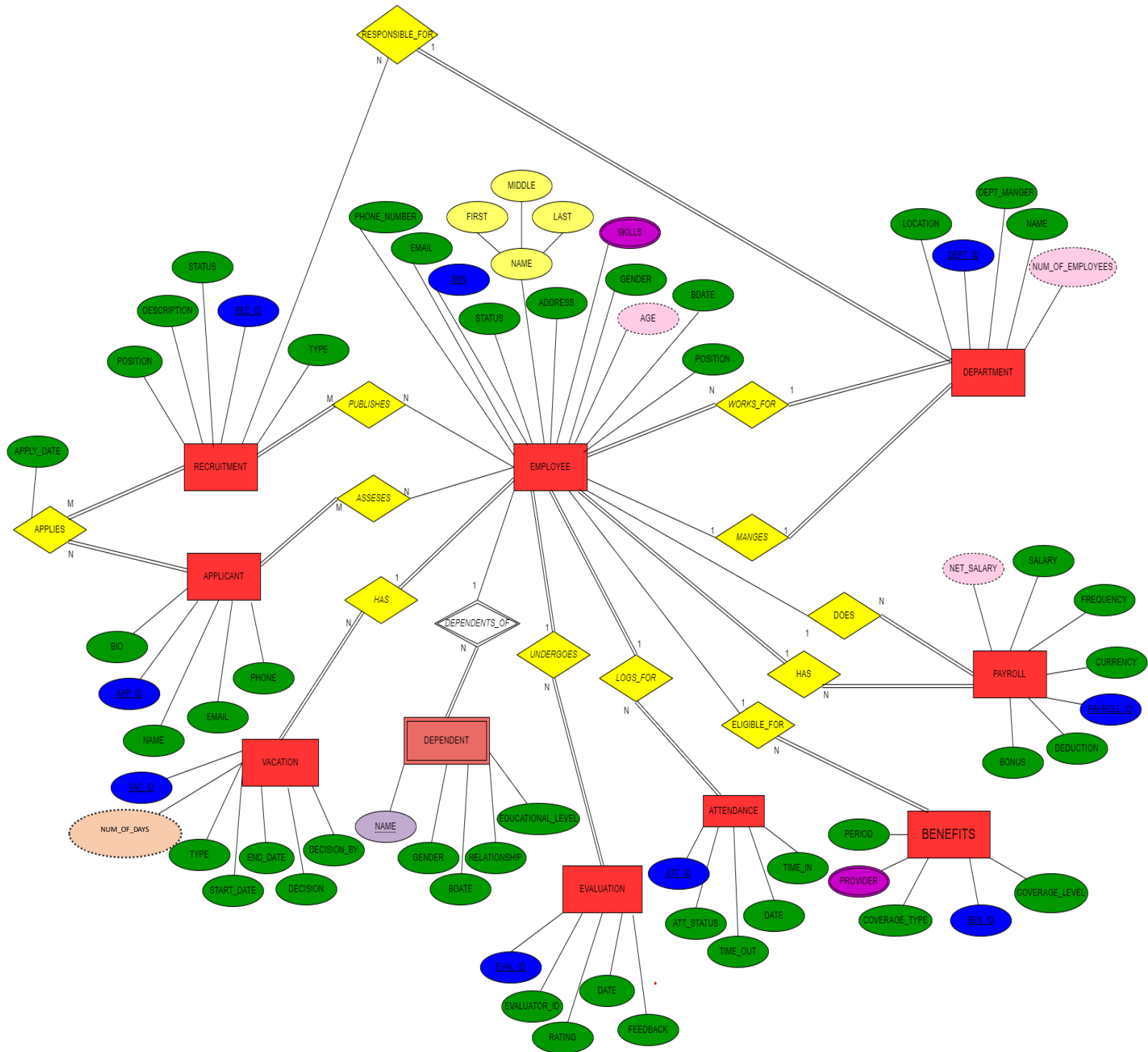
This material is intended for informational purposes only about the database system of CraftersHR and how it works, and Data Crafters makes no warranties regarding its accuracy or completeness. The advice and strategies contained in this material may not be suitable for every situation, and Data Crafters disclaims all liability for any loss or damages.

### III. ER Legend

Name	Legend
Primary Key	
Attribute	
Multivalued Attribute	
Entity	
Weak Entity	
Partial Key	
Relationship	
Identifying Relationship	

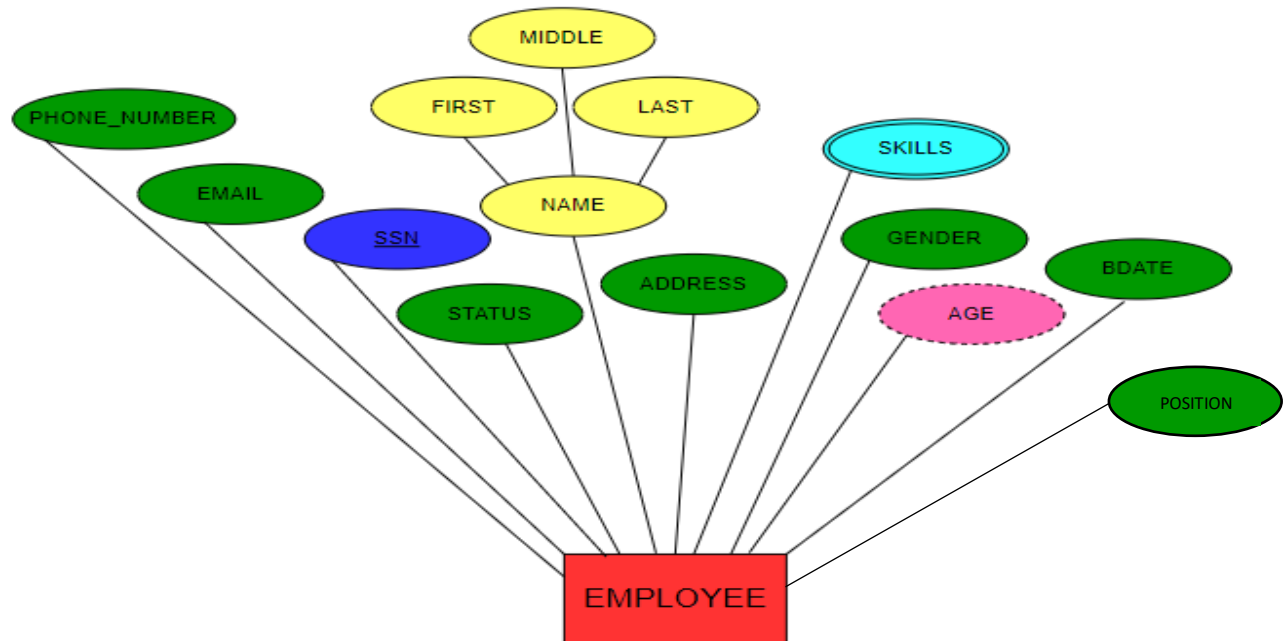
<b>Total Participation</b>	=====
<b>Partial Participation</b>	=====

## IV. ER Diagram



## V. Entities and their attributes

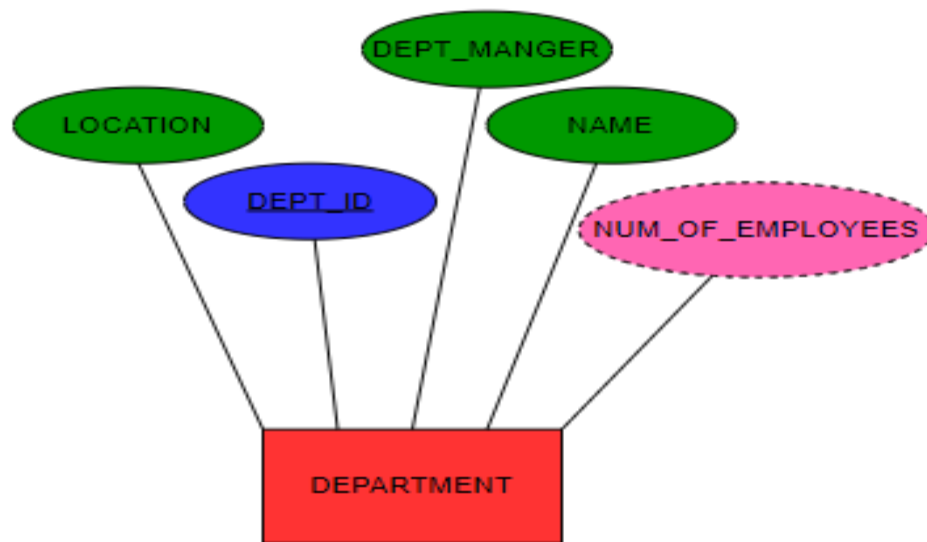
- 1- **EMPLOYEE:** A strong entity that represents individuals employed by the company, storing essential information like their personal details and contact information.



- **SSN (Social Security Number):** This serves as the unique identifier and **key** attribute for each employee, ensuring each record is distinct.
- **NAME:** This attribute is **composite**, meaning it's comprised of multiple sub-attributes:
  - **First:** Denotes the employee's given or first name.
  - **Middle:** Represents the employee's middle name.
  - **Last:** Indicates the employee's surname or family name.
- **ADDRESS:** Refers to the physical address where the employee resides or can be reached.
- **EMAIL:** This attribute stores the employee's email address for communication purposes.
- **PHONE\_NUMBER:** Stores the employee's contact number, typically a mobile or landline number.
- **GENDER:** Indicates the gender identity of the employee.
- **BDATE (Birth Date):** Records the employee's date of birth, providing insight into their age.
- **AGE:** **Derived** attribute calculated from the employee's birth date. It provides the current age of the employee, which can be automatically computed based on the birth date.
- **STATUS:** Denotes the current employment status of the employee (e.g., full-time, part-time, contractor, etc.).

- **SKILLS:** This attribute is **multivalued**, meaning it can contain multiple values. It stores the skills possessed by the employee, which can be crucial for matching employees with suitable job roles or projects.
- **POSITION:** This attribute describes the specific role or job title that the employee holds.

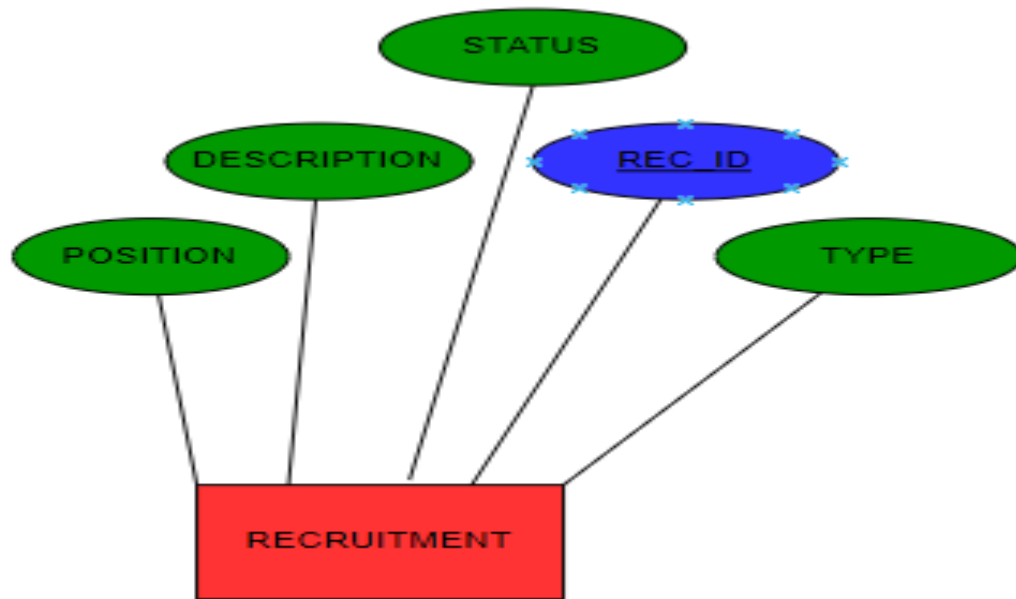
2- **DEPARTMENT:** A strong entity within the organizational structure, representing distinct departments within the company and storing essential information such as location and departmental manager.



- **DEPT\_ID (Department ID):** Serves as the unique identifier and **key** attribute for each department, ensuring each record is distinct.
- **LOCATION:** Refers to the physical location or site where the department is situated within the organization.
- **NAME:** Denotes the name or title of the department, providing clarity on its function or purpose.
- **DEPT\_MANAGER:** Indicates the individual responsible for overseeing and managing the department's operations and personnel.
- **NUM\_OF\_EMPLOYEES:** A **derived** attribute calculated based on the number of employees assigned to the department. It provides insight into the department's workforce size, aiding in resource allocation and management.

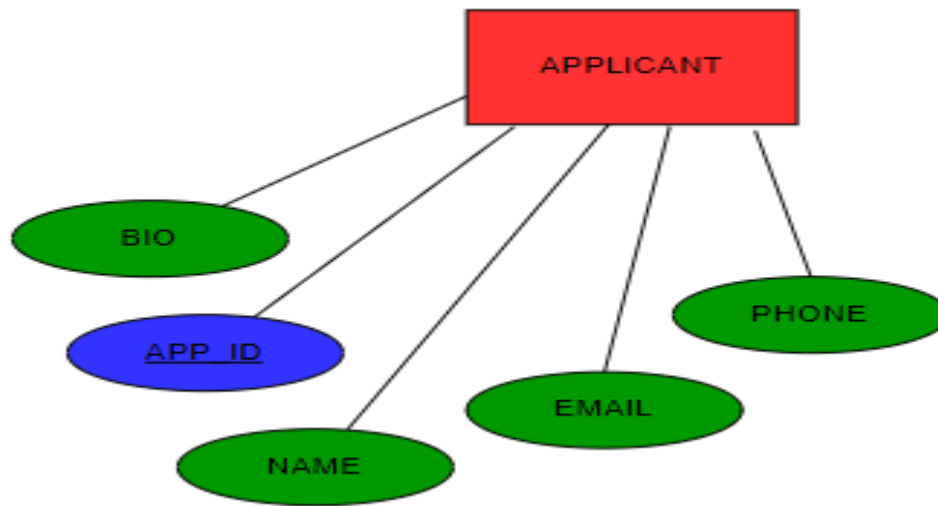


- 3- **RECRUITMENT:** A strong entity in the HR system responsible for managing recruitment processes and vacancies within the organization.



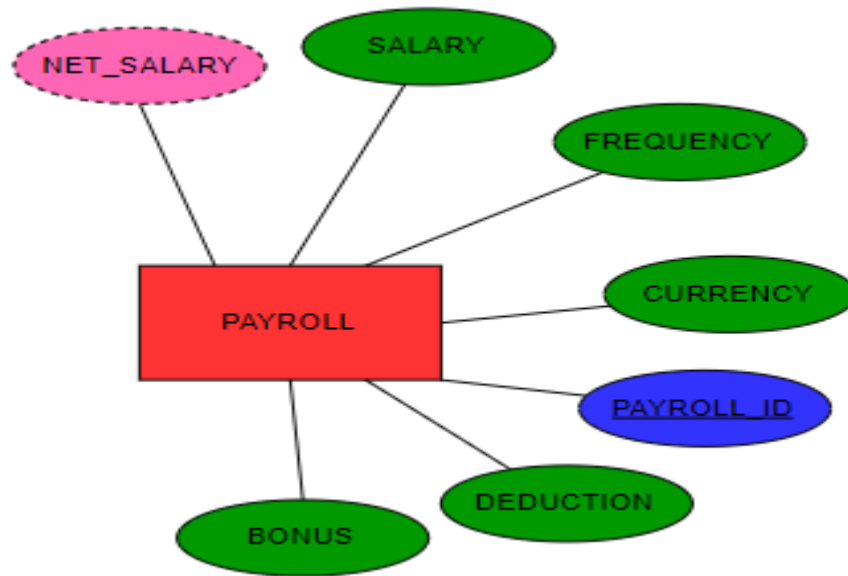
- **REC\_ID (Recruitment ID):** Serves as the unique identifier and **key** attribute for each recruitment record, ensuring distinct tracking of recruitment activities.
- **POSITION:** Specifies the job position or title for which recruitment is being conducted, providing clarity on the role's responsibilities and requirements.
- **DESCRIPTION:** Describes the details and specifications of the job position, including duties, qualifications, and expectations.
- **STATUS:** Indicates the current status of the recruitment process for the particular position (e.g., open, closed, in progress), facilitating tracking and monitoring of recruitment activities.
- **TYPE:** Specifies the nature of the job position (e.g., internship, full-time, part-time), categorizing positions based on employment type to streamline recruitment efforts and target suitable candidates.

- 4- **APPLICANT:** A robust entity within the HR system, representing individuals who have applied for job positions within the organization.



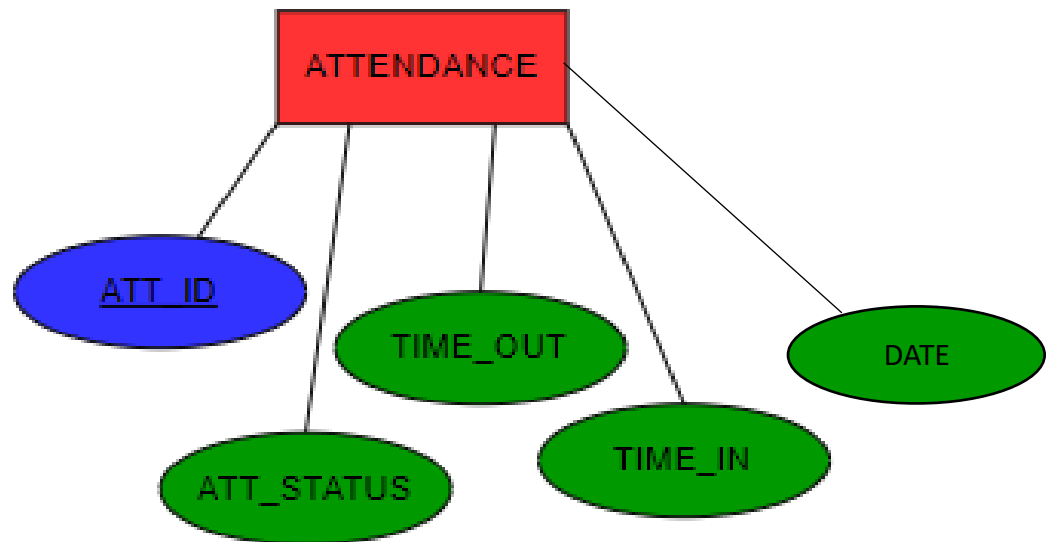
- **APP\_ID (Applicant ID):** Unique identifier and **key** attribute for each applicant, ensuring distinct tracking of application records.
- **NAME:** Name of the applicant, providing identification and contact information.
- **BIO:** Brief biography or profile of the applicant, including relevant background information, resume, skills, and experiences.
- **EMAIL:** Email address of the applicant for communication purposes and correspondence regarding the application.
- **PHONE:** Contact phone number of the applicant, facilitating communication and follow-up regarding the application process.

- 5- **PAYROLL**: A foundational entity within the HR system, managing the financial compensation and payments for employees within the organization.



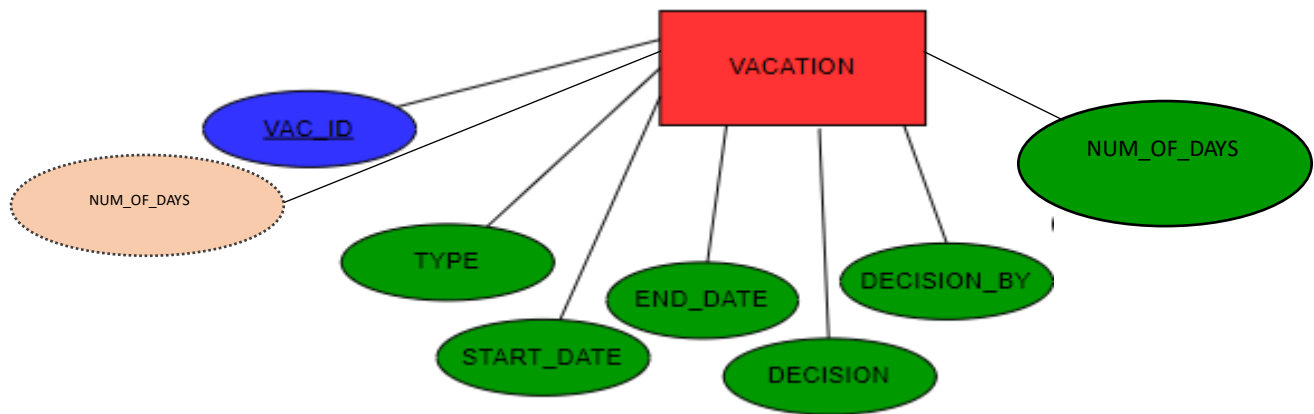
- **PAYROLL\_ID (Payroll ID)**: Unique identifier and **key** attribute for each payroll record, ensuring distinct tracking of payroll transactions.
- **CURRENCY**: Denotes the currency used for salary payments, providing clarity on the monetary unit.
- **FREQUENCY**: Specifies the frequency of payroll payments (e.g., monthly, bi-weekly, weekly), indicating how often employees are paid.
- **DEDUCTION**: Represents any deduction or withholding from the employee's salary, such as taxes, insurance premiums, or retirement contributions.
- **BONUS**: Indicates any additional bonus or incentive provided to employees, beyond their regular salary.
- **SALARY**: Represents the base salary or wage amount paid to employees for their work.
- **NET SALARY (Derived)**: **Derived** attribute representing the employee's take-home pay after deductions and bonuses. It provides the final amount received by the employee, facilitating transparency in payroll calculations.

- 6- **ATTENDANCE:** A core entity in the HR system, tracking employees' attendance and time-related information within the organization.



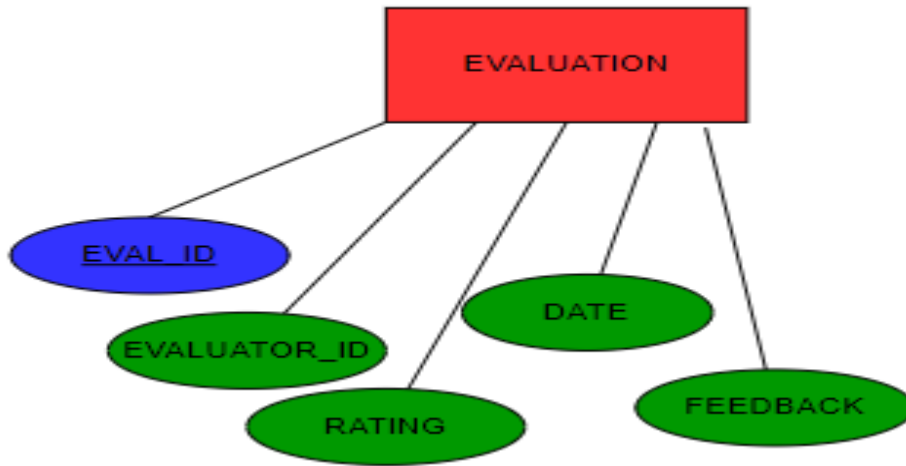
- **ATT\_ID (Attendance ID):** Unique identifier and **key** attribute for each attendance record, ensuring distinct tracking of attendance data.
- **ATT\_STATUS:** Indicates the attendance status of the employee for a specific time period (e.g., present, absent, late), providing insight into their punctuality and attendance behavior.
- **TIME\_OUT:** Records the time when the employee clocked out or ended their work shift, helping to monitor working hours and schedule adherence.
- **TIME\_IN:** Records the time when the employee clocked in or started their work shift, enabling tracking of arrival times and attendance patterns.
- **DATE:** Records the exact date when the employee checked in to their work shift.

- 7- **VACATION:** A strong entity within the HR system, managing employee vacation requests and approvals within the organization.



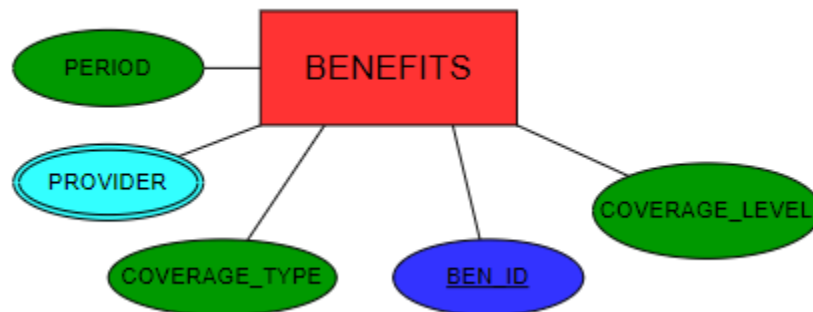
- **VAC\_ID (Vacation ID):** Unique identifier and **key** attribute for each vacation request, ensuring distinct tracking of vacation records.
- **TYPE:** Specifies the type of vacation requested by the employee (e.g., annual leave, sick leave, unpaid leave), categorizing different types of time-off.
- **START\_DATE:** Indicates the start date of the employee's vacation period, providing clarity on when the leave begins.
- **END\_DATE:** Indicates the end date of the employee's vacation period, specifying the duration of the leave.
- **DECISION:** Indicates the decision made on the vacation request (e.g., approved, denied, pending), informing the employee of the status of their request.
- **DECISION\_BY:** Specifies the individual or authority responsible for making decisions regarding the vacation request.
- **NUM\_OF\_DAYS (Derived):** Determines the number of days of the vacation taken by the employee, which clarifies the duration of it.

- 8- **EVALUATION:** An essential entity in the HR system, managing employee performance evaluations and feedback within the organization.



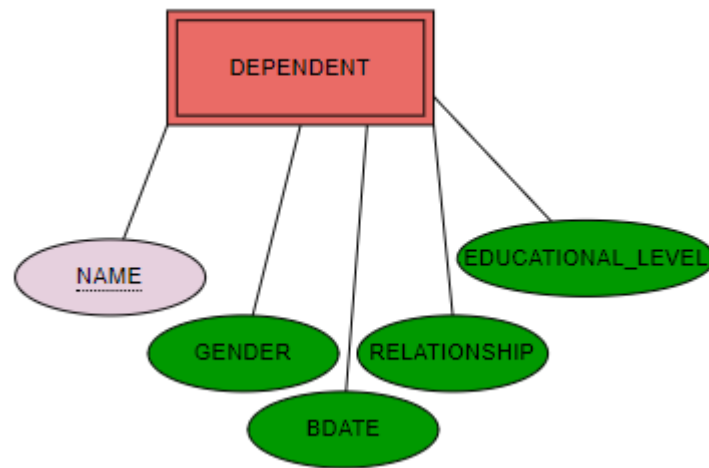
- **EVAL\_ID (Evaluation ID):** Unique identifier and **key** attribute for each evaluation record, ensuring distinct tracking of evaluation instances.
- **EVALUATOR\_ID:** Identifies the individual responsible for conducting the evaluation, providing clarity on the evaluator's role.
- **DATE:** Specifies the date when the evaluation was conducted, indicating the timing of the performance assessment.
- **RATING:** Represents the numerical rating assigned to the employee's performance, reflecting their achievements and areas for improvement.
- **FEEDBACK:** Contains feedback or comments provided to the employee during the evaluation process, offering insights and guidance for professional development.

- 9- **BENEFITS:** A strong entity within the HR system, managing employee benefits and coverage options provided by the organization.



- **BEN\_ID (Benefit ID):** Unique identifier and **key** attribute for each benefit record, ensuring distinct tracking of benefit offerings.
- **COVERAGE\_TYPE:** Specifies the type of coverage offered by the benefit (e.g., health insurance, dental insurance, retirement plan), categorizing benefits based on their purpose.
- **COVERAGE\_LEVEL:** Indicates the level or extent of coverage provided by the benefit (e.g., individual, family), specifying the scope of the benefit.
- **PROVIDER (Multivalued):** Stores information about the provider(s) offering the benefit, which can include insurance companies, investment firms, or other service providers.
- **PERIOD:** Specifies the period or duration for which the benefit coverage is effective, indicating the timeframe during which employees can utilize the benefit.

**10- DEPENDENT:** A weak entity related to the employee entity, representing individuals who are dependent on the employee for benefits or support.

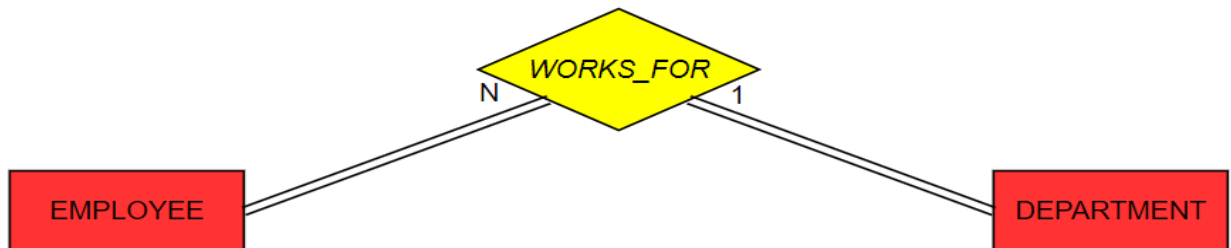


- **NAME (Partial Key):** The name of the dependent, serving as part of the **key** attribute for identifying the dependent within the system.
- **GENDER:** Specifies the gender identity of the dependent.
- **BDATE (Birth Date):** Indicates the date of birth of the dependent, providing insight into their age.
- **RELATIONSHIP:** Describes the relationship between the dependent and the employee (e.g., spouse, child, parent).
- **EDUCATIONAL\_LEVEL:** Specifies the educational level or attainment of the dependent, if applicable.



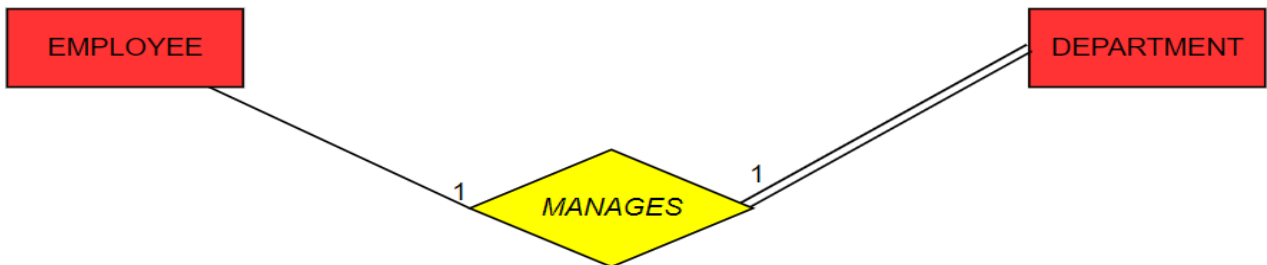
## VI. Relationships

### 1. WORKS\_FOR:



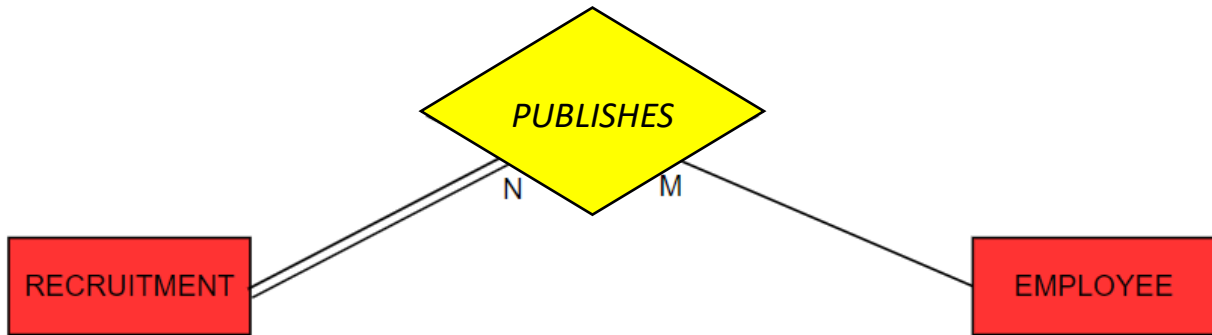
The relationship "WORKS\_FOR" between EMPLOYEE and DEPARTMENT signifies that each employee works for one department, while each department can have many employees. This is a one-to-many relationship. In terms of participation, it is full on both sides, meaning that all employees must work for a department (full participation of employees), and every department must have at least one employee working for it (full participation of departments). It facilitates organization and management within the HR system by establishing clear connections between employees and their respective departments.

### 2. MANAGES:



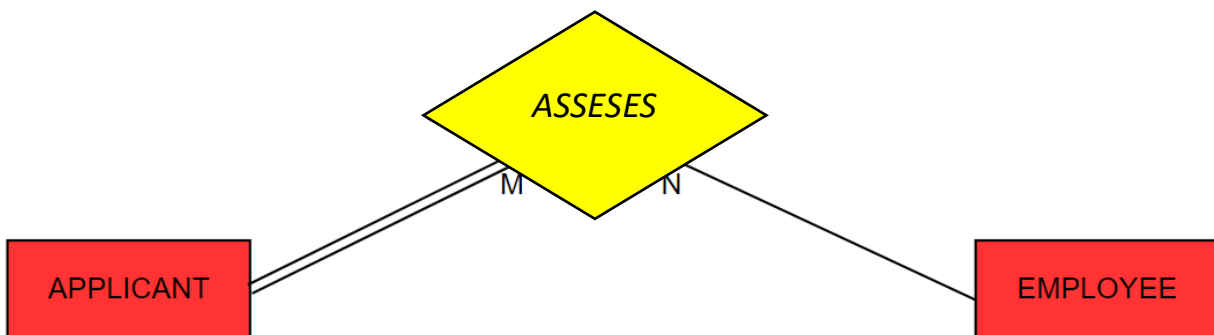
The relationship "MANAGES" between EMPLOYEE and DEPARTMENT signifies that each department is managed by one employee, while an employee may or may not manage a department. This is a one-to-one relationship. In terms of participation, it is partial on the EMPLOYEE side, meaning that not every employee necessarily manages a department (partial participation), but it is full on the DEPARTMENT side, indicating that every department must have a manager (full participation). It facilitates hierarchical organization and management within the HR system by establishing clear leadership roles within each department.

### 3. PUBLISHES:



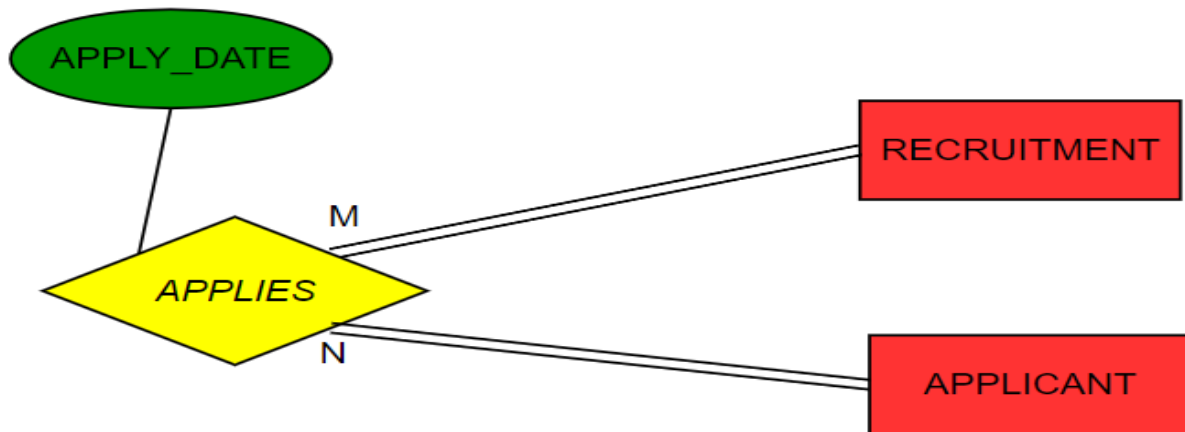
The relationship "PUBLISHES" between EMPLOYEE and RECRUITMENT signifies that an employee may or may not be involved in publishing a recruitment announcement, while each recruitment announcement must be published by at least one employee. This is a many-to-many relationship. In terms of participation, it is partial on the EMPLOYEE side, indicating that not every employee is involved in publishing recruitment announcements (partial participation), but it is full on the RECRUITMENT side, suggesting that every recruitment announcement must be published by at least one employee (full participation). It facilitates the management of recruitment processes and ensures that each announcement has a designated publisher while allowing multiple employees to be involved in publishing recruitment announcements.

### 4. ASSESSES:



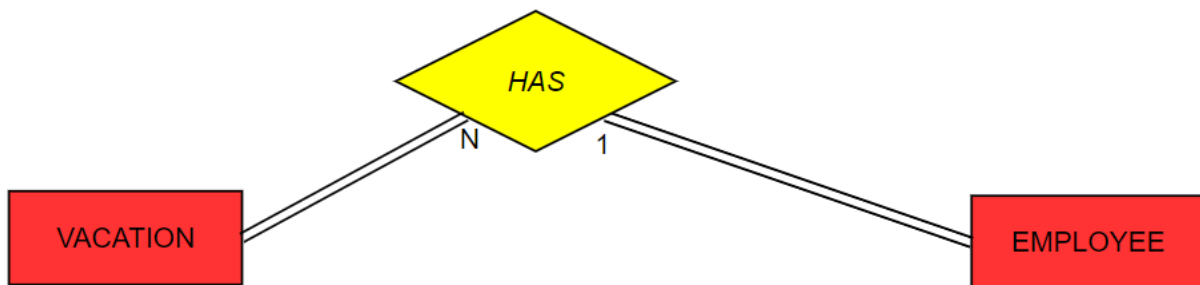
The relationship "ASSESES" between EMPLOYEE and APPLICANT signifies that an employee may or may not be involved in assessing job applicants, while each job applicant must be assessed by at least one employee. This is a many-to-many relationship. In terms of participation, it is partial on the EMPLOYEE side, indicating that not every employee is involved in assessing job applicants (partial participation), but it is full on the APPLICANT side, suggesting that every job applicant must be assessed by at least one employee (full participation). It facilitates the evaluation process during recruitment and ensures that each applicant receives assessment while allowing multiple employees to be involved in evaluating candidates.

## 5. APPLIES:



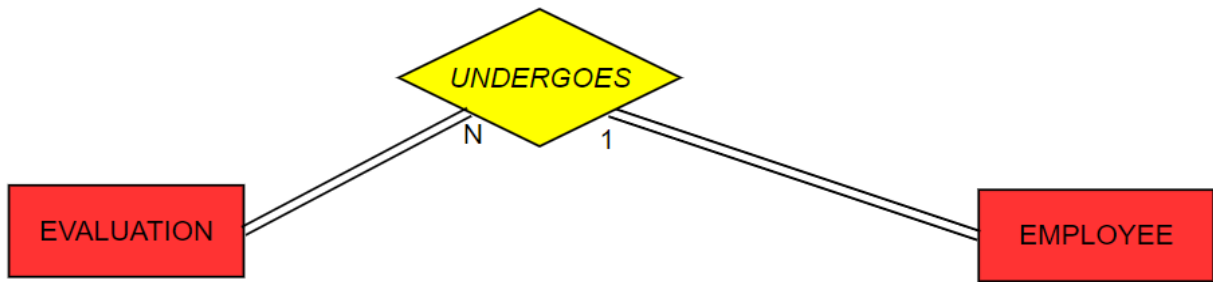
The relationship "APPLIES" between RECRUITMENT and APPLICANT signifies that each recruitment process can have many applicants, and each applicant can apply to multiple recruitment processes. This is a many-to-many relationship. In terms of participation, it is full on both sides, meaning that every recruitment process must have at least one applicant (full participation of recruitment), and every applicant must apply to at least one recruitment process (full participation of applicants). Additionally, there is a relationship attribute called "APPLY\_DATE," which represents the date when an applicant applied to a specific recruitment process. It allows the HR system to manage the application process efficiently while recording important details such as application dates.

## 6. HAS:



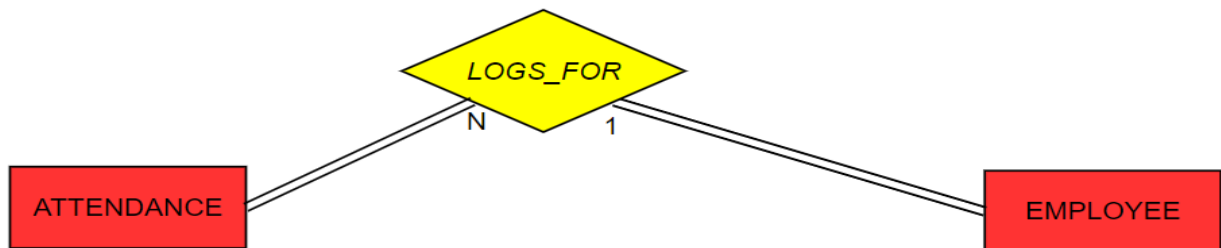
The relationship "HAS" between EMPLOYEE and VACATION indicates that each employee may have multiple vacation records, while each vacation record belongs to exactly one employee. This is a one-to-many relationship. In terms of participation, it is total on both sides, meaning that every employee must have at least one vacation record (total participation of employees), and every vacation record must belong to exactly one employee (total participation of vacation records). This relationship facilitates the management of vacation records for employees within the HR system, ensuring that each employee's vacations are accurately recorded and tracked. It allows for efficient monitoring of employee time-off and scheduling.

## 7. UNDERGOES:



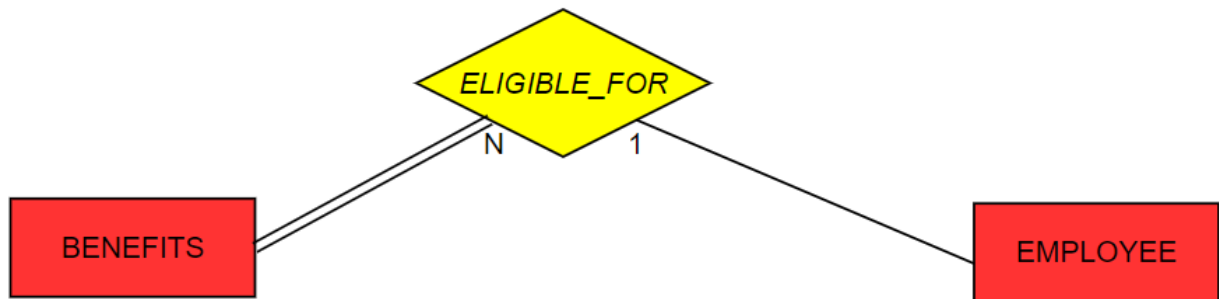
The relationship "UNDERGOES" between EMPLOYEE and EVALUATION signifies that each employee undergoes multiple evaluations, while each evaluation is conducted for exactly one employee. This is a one-to-many relationship. In terms of participation, it is total on both sides, meaning that every employee must undergo at least one evaluation (total participation of employees), and every evaluation must be conducted for exactly one employee (total participation of evaluations). This relationship facilitates the management of employee performance evaluations within the HR system, ensuring that each employee's performance is assessed and documented accurately. It allows for the tracking of evaluation history and employee progress over time.

## 8. LOGS\_FOR:



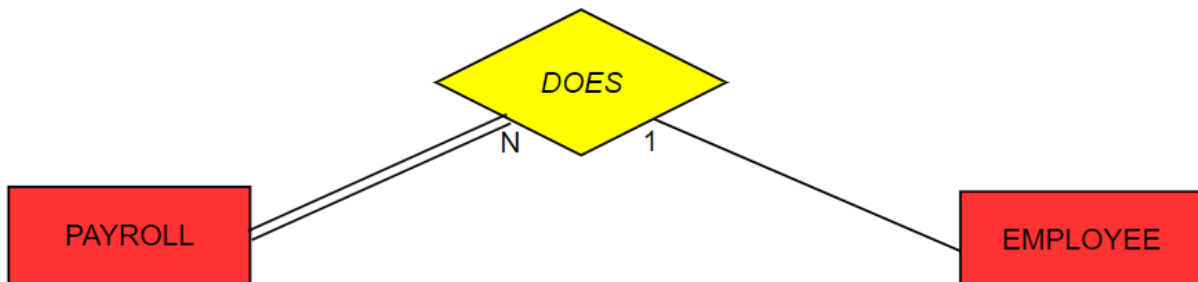
The relationship "LOGS\_FOR" between EMPLOYEE and ATTENDANCE denotes that each employee logs multiple attendance records, while each attendance record is logged by exactly one employee. This represents a one-to-many relationship. Regarding participation, it is total on both sides, signifying that every employee must have at least one attendance record (total participation of employees), and every attendance record must be logged by exactly one employee (total participation of attendance records). This relationship facilitates the tracking of employee attendance within the HR system, ensuring that each employee's attendance is accurately recorded and monitored. It enables HR personnel to manage attendance data efficiently and track employee attendance patterns over time.

## 9. ELIGIBLE\_FOR:



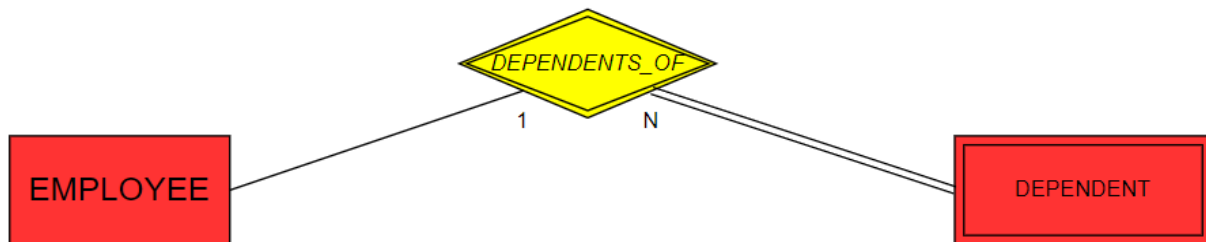
The relationship "ELIGIBLE\_FOR" between EMPLOYEE and BENEFITS indicates that each employee may be eligible for multiple benefits, while each benefit is associated with exactly one employee. This represents a one-to-many relationship. In terms of participation, it is total on the BENEFITS side, meaning that every benefit must be associated with at least one employee (total participation of benefits), and it is partial on the EMPLOYEE side, suggesting that not every employee may be eligible for benefits (partial participation of employees). This relationship facilitates the management of employee benefits within the HR system, ensuring that each eligible employee is associated with relevant benefits. It allows HR administrators to track benefit eligibility for employees and manage benefit offerings effectively.

## 10. DOES:



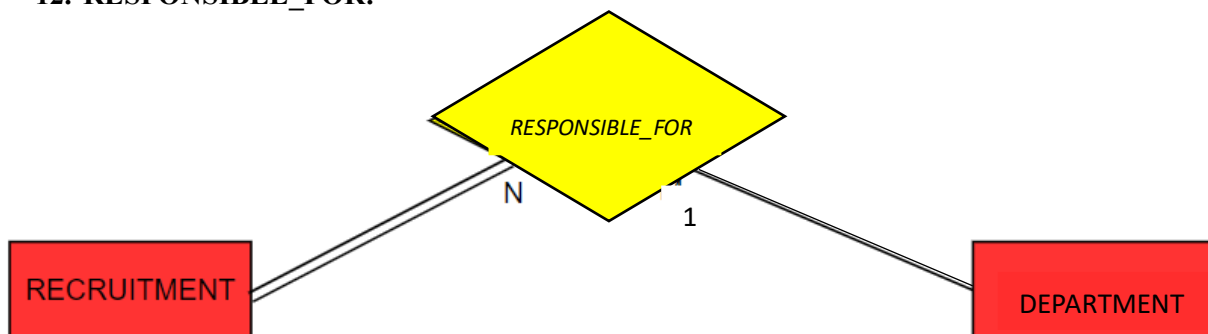
The relationship "DOES" between EMPLOYEE and PAYROLL denotes that each employee may have multiple payroll records, while each payroll record is associated with exactly one employee. This represents a one-to-many relationship. In terms of participation, it is total on the PAYROLL side, indicating that every payroll record must be associated with at least one employee (total participation of payroll records), and it is partial on the EMPLOYEE side, suggesting that not every employee may have payroll records (partial participation of employees). This relationship facilitates the management of payroll records within the HR system, ensuring that each employee's payroll information is accurately recorded and managed. It allows HR personnel to track and manage employee compensation efficiently.

## 11. DEPENDENTS\_OF:



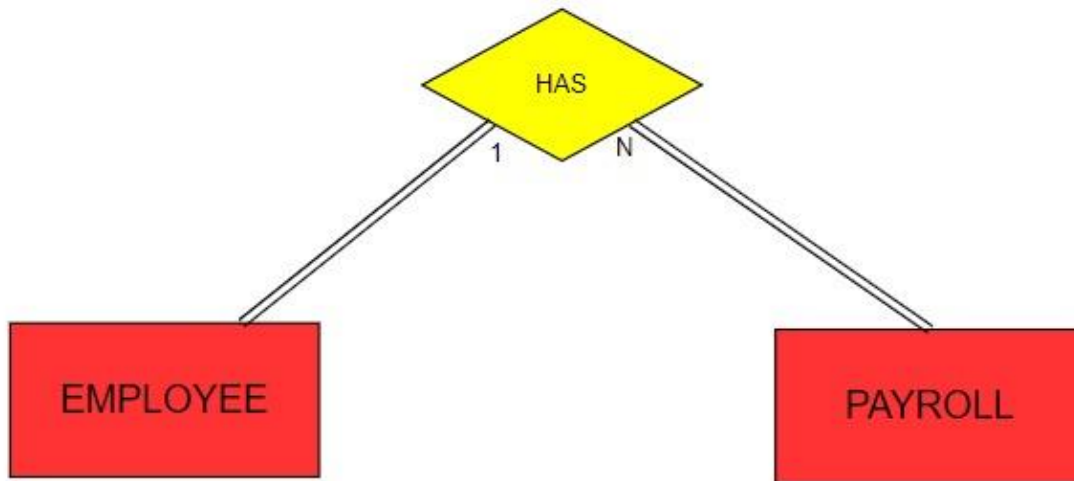
The identifying relationship "DEPENDENTS\_OF" between EMPLOYEE and the weak entity DEPENDENT signifies that each employee may have multiple dependents, while each dependent is associated with exactly one employee. This represents a one-to-many relationship. In terms of participation, it is total on the DEPENDENT side, indicating that every dependent must be associated with exactly one employee (total participation of dependents), and it is partial on the EMPLOYEE side, suggesting that not every employee may have dependents (partial participation of employees). This relationship facilitates the management of dependent information within the HR system, ensuring that each dependent is properly linked to their corresponding employee. It allows HR personnel to track and manage employee-dependent relationships efficiently.

## 12. RESPONSIBLE\_FOR:



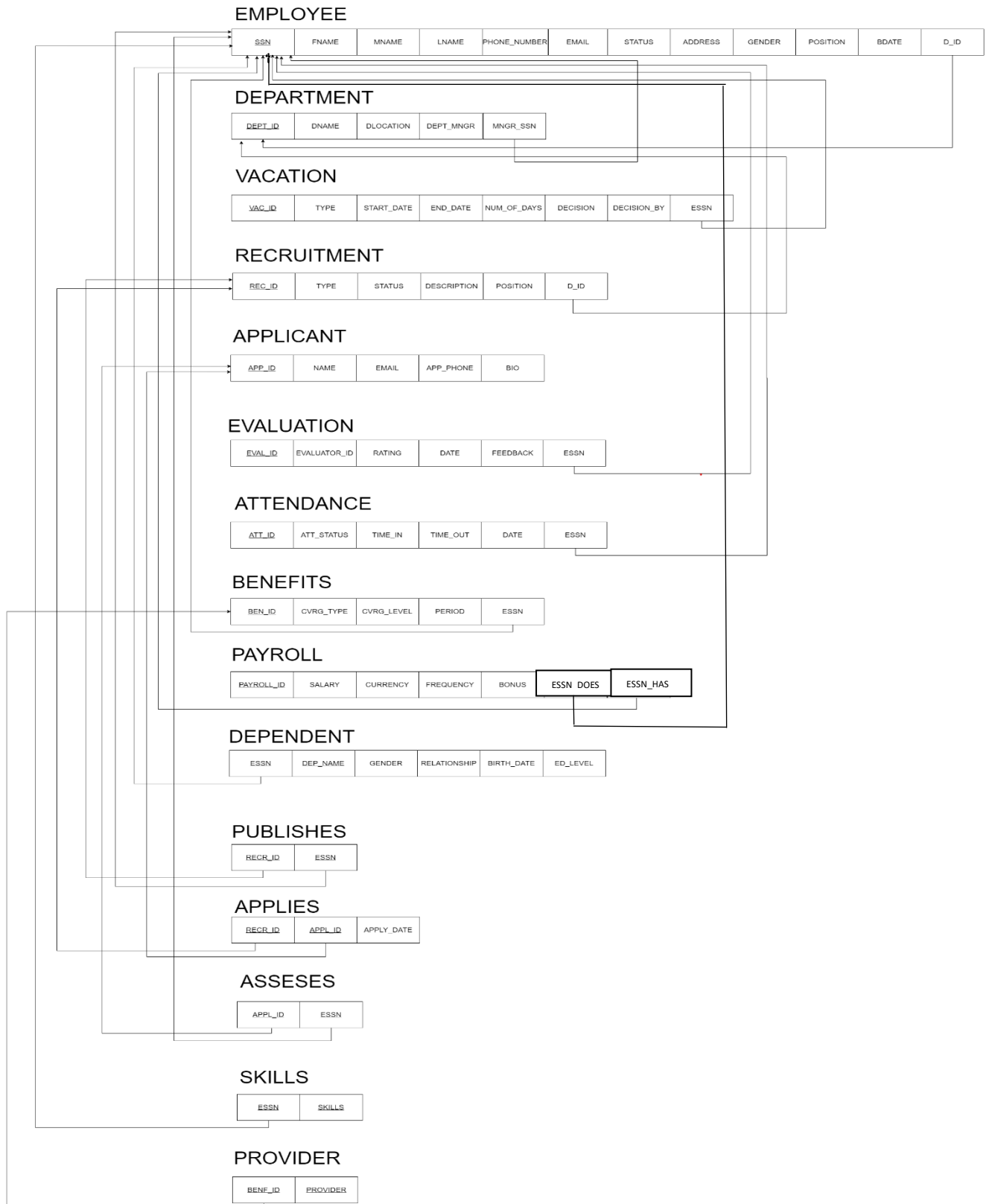
The relationship "RESPONSIBLE\_FOR" between DEPARTMENT and RECRUITMENT denotes that each department may oversee multiple recruitment efforts, while each recruitment effort is exclusively overseen by one department. This forms a one-to-many relationship. In terms of participation, it is total on both sides, signifying that every department must take charge of at least one recruitment initiative (total participation of departments), and every recruitment effort must have a designated overseeing department (total participation of recruitment efforts). It enables efficient coordination and oversight of recruitment efforts, empowering departments to effectively manage their talent acquisition strategies and contribute to organizational success.

### 13. HAS:



In a one-to-many relationship between an employee and payroll, denoted by the relationship name "has," each employee can have multiple payroll records, but each payroll record belongs to only one employee. This setup signifies that an employee can receive multiple payroll payments over time, while each payroll entry corresponds uniquely to an individual employee. With partial participation on both sides, it means that not every employee necessarily has a payroll record, and similarly, not every payroll record must be associated with an employee, allowing for flexibility in the data model.

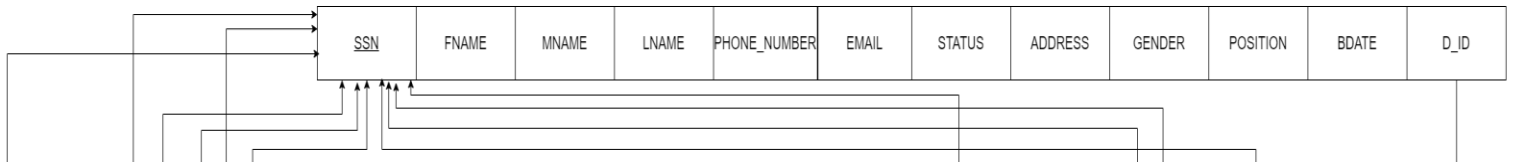
## VII. Mapping:



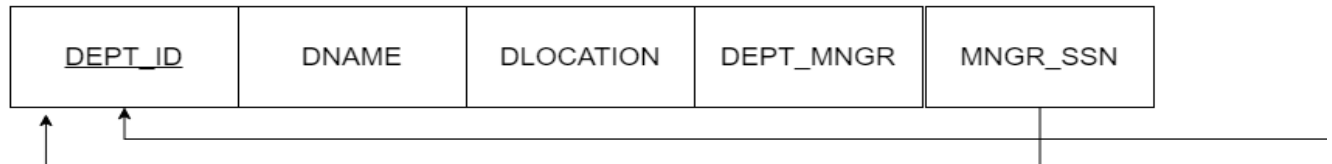


i. Mapping of strong entities:

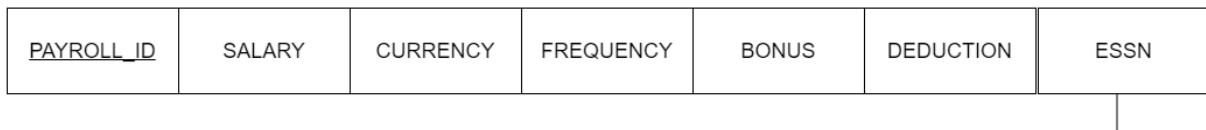
## EMPLOYEE



## DEPARTMENT



## PAYROLL



## BENEFITS



## EVALUATION



# APPLICANT




<u>APP_ID</u>	NAME	EMAIL	APP_PHONE	BIO
---------------	------	-------	-----------	-----

# RECRUITMENT




<u>REC_ID</u>	TYPE	STATUS	DESCRIPTION	POSITION	D_ID
---------------	------	--------	-------------	----------	------




# VACATION

<u>VAC_ID</u>	TYPE	START_DATE	END_DATE	NUM_OF_DAYS	DECISION	DECISION_BY	ESSN
---------------	------	------------	----------	-------------	----------	-------------	------



# ATTENDANCE

<u>ATT_ID</u>	ATT_STATUS	TIME_IN	TIME_OUT	DATE	ESSN
---------------	------------	---------	----------	------	------



For each strong entity, we created a corresponding table (relation) in the relational schema. Where the table name should match the name of the strong entity and each table will have columns (attributes) that correspond to the attributes of the strong entity. The resulting table includes the key attribute which will be named as primary key (underlined), and also all composite and simple attributes. Above is an example of a mapping of a strong entity which is EMPLOYEE.

## ii. Mapping of weak entity:

### DEPENDENT

<u>ESSN</u>	<u>DEP_NAME</u>	GENDER	RELATIONSHIP	BIRTH_DATE	ED_LEVEL
-------------	-----------------	--------	--------------	------------	----------

For each weak entity, we created a corresponding table (relation) in the relational schema. Where the table name should match the name of the weak entity and each table will have columns (attributes) that correspond to the attributes of the weak entity and we included all the attributes. We created a foreign key in the DEPENDENT table which is ESSN that references the primary key of EMPLOYEE which is SSN. This foreign key establishes the relationship between the weak entity and its owner. Now both the foreign key ESSN and the partial key of the weak entity DEP\_NAME form the primary key of the weak relation {ESSN, DEP\_NAME}.

## iii. Mapping of one-to-one relationship:

### EMPLOYEE

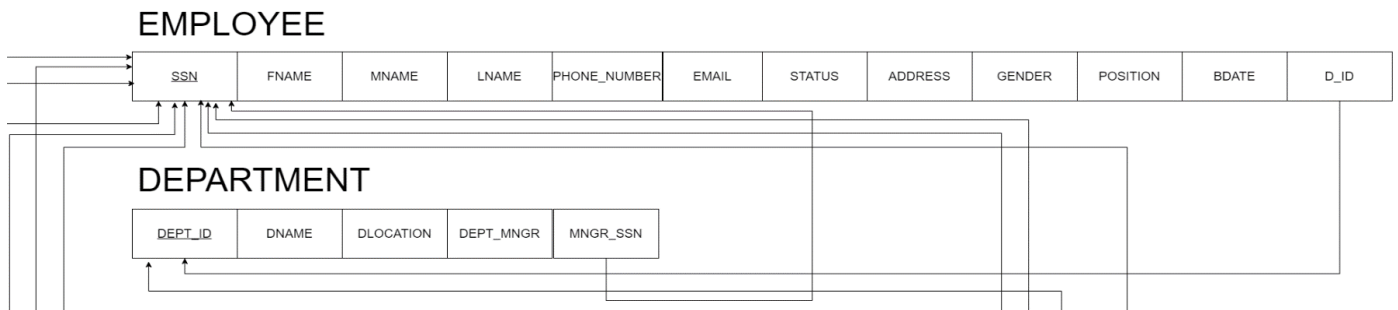
<u>SSN</u>	FNAME	MNAME	LNAME	PHONE_NUMBER	EMAIL	STATUS	ADDRESS	GENDER	POSITION	BDATE	D_ID
------------	-------	-------	-------	--------------	-------	--------	---------	--------	----------	-------	------

### DEPARTMENT

<u>DEPT_ID</u>	DNAME	DLOCATION	DEPT_MNGR	MNGR_SSN
----------------	-------	-----------	-----------	----------

For the one-to-one relationship we have which is MANAGES, we used the foreign key approach and in the foreign key approach, we choose one of the tables (let's call it S) to be the parent table (better to be the one with full participation) then we include a foreign key in the parent table (S) that references the primary key of the other table (T). The foreign key establishes the relationship between the two tables. This approach ensures data integrity and consistency. Here we included MNGR\_SSN as a foreign key in DEPARTMENT and it is referencing the primary key of EMPLOYEE which is SSN.

iv. Mapping of one-to-many relationships:



## ATTENDANCE

<u>ATT_ID</u>	ATT_STATUS	TIME_IN	TIME_OUT	DATE	ESSN
---------------	------------	---------	----------	------	------

## EVALUATION

<u>EVAL_ID</u>	EVALUATOR_ID	RATING	DATE	FEEDBACK	ESSN
----------------	--------------	--------	------	----------	------

## VACATION

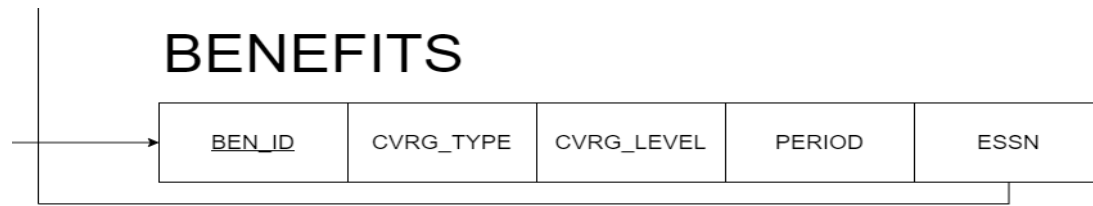
<u>VAC_ID</u>	TYPE	START_DATE	END_DATE	NUM_OF_DAYS	DECISION	DECISION_BY	ESSN
---------------	------	------------	----------	-------------	----------	-------------	------

## PAYROLL

<u>PAYROLL_ID</u>	SALARY	CURRENCY	FREQUENCY	BONUS	DEDUCTION	ESSN
-------------------	--------	----------	-----------	-------	-----------	------

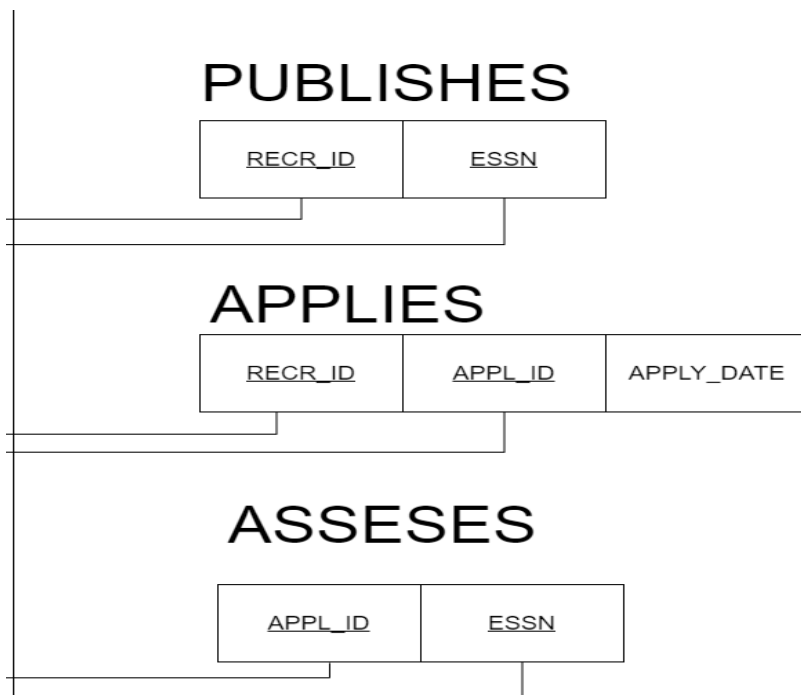
## RECRUITMENT

<u>REC_ID</u>	TYPE	STATUS	DESCRIPTION	POSITION	D_ID
---------------	------	--------	-------------	----------	------



For the one-to-many relationships we have, we used something similar to foreign key approach but we chose the table on the N side of the relationship to be the parent table and we included a foreign key in it that references the primary key of the table on the 1 side of the relationship. As an example, we have the *ELIGIBLE\_FOR* as 1-to-many relationship where we took the primary key of the EMPLOYEE which is SSN and included it as foreign key in BENEFITS which is ESSN as it references SSN.

#### v. Mapping of many-to-many relationships:



For the many-to-many relationships, we created new relation to represent it and this is a relationship relation. We included as foreign key attributes in this relation the primary keys of the relations that represent the participating entity types; as their combination will form the primary

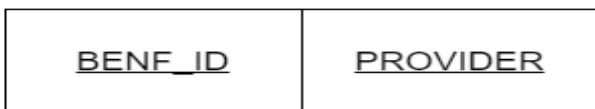
key of the new relation. Also we included any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of this relation. As an example, we have the *PUBLISHES* relationship, where we took the primary keys of both sides of this relationship, SSN from EMPLOYEE and RECR\_ID from RECRUITMENT and included them as foreign keys in the new relation named ESSN and RECR\_ID respectively. Now both these 2 foreign keys that references the primary ones form the primary key of the new relation {RECR\_ID, ESSN}.

vi. Mapping of multivalued attributes:

## SKILLS



## PROVIDER



For the multivalued attributes, we created a new relation to represent it and what we did is that we included the multivalued attribute itself in the relation along with including a foreign key that references the primary key of the entity that this multivalued attribute belongs to. Then these two will form together the primary key of the new relation. As an example, we have SKILLS as multivalued attribute in the EMPLOYEE entity, so we included SKILLS itself and SSN as ESSN in the new relation where both make the primary key of it {ESSN, SKILL}.

vii. **Summary:**

- **First Step:** Mapping of the strong entities where we created a relation for each one containing its primary key, simple attributes and composite attributes. (ex: EMPLOYEE, DEPARTMENT, RECRUITMENT, PAYROLL, BENEFITS, ATTENDANCE, EVALUATION, VACATION, APPLICANT)
- **Second Step:** Mapping of the weak entity where we created a relation including its partial key and simple attributes. (ex: DEPENDENT)
- **Third Step:** Mapping of the one-to-one relationships using the foreign key approach where we took the primary key from the partially participating entity and setting it as a foreign key in the fully participating one. (ex: MANAGES)
- **Fourth Step:** Mapping of the one-to-many where we took the primary key from the entity with cardinality ratio of many and setting it as a foreign key in the other entity. (ex: HAS, DOES, RESPONSIBLE\_FOR, WORKS\_FOR, ELIGIBLE\_FOR, LOGS\_FOR, UNDERGOES, HAS)
- **Fifth Step:** Mapping of the many-to-many where we created a new relation and added the primary key of each corresponding entity in which both form the new primary key for the relation, also we added relationship attributes if any. (ex: PUBLISHES, ASSESSES, APPLIES)
- **Sixth Step:** Mapping of the multivalued attributes where we created a new relation for each one and added the primary key of the owner entity with the attribute itself for them both to form together the primary key of the relation. (ex: SKILLS, PROVIDER)

## VIII. Creation of Tables:

The CREATE TABLE query is used to create a new table in the database. It specifies the structure of the table by defining its columns along with their data types. Additionally, constraints such as primary keys can be defined to enforce data integrity rules. The query begins with the CREATE TABLE statement followed by the name of the table being created. Inside the parentheses, each column of the table is defined with its name and data type. Constraints such as primary keys are also specified within the parentheses. Overall, the CREATE TABLE query establishes the blueprint for a new table in the database, defining its structure and any constraints necessary for data integrity.

- Table EMPLOYEE:

```
CREATE TABLE EMPLOYEE (  
SSN NUMBER NOT NULL PRIMARY KEY,  
EMAIL VARCHAR2(50),  
PH_NUM VARCHAR2(50),  
STATUS VARCHAR2(50),  
ADDRESS VARCHAR2(50),  
FNAME VARCHAR2(50),  
MNAME VARCHAR2(50),  
LNAME VARCHAR2(50),  
GENDER VARCHAR2(50),  
BDATE VARCHAR2(50),  
POSITION VARCHAR2(50),  
D_ID NUMBER,  
FOREIGN KEY (D_ID) REFERENCES department (DEPT_ID));
```

Column Name	Data Type	Nullable	Default	Primary Key
SSN	NUMBER	No	-	1
EMAIL	VARCHAR2(50)	Yes	-	-
PH_NUM	VARCHAR2(50)	Yes	-	-
STATUS	VARCHAR2(50)	Yes	-	-
ADDRESS	VARCHAR2(50)	Yes	-	-
FNAME	VARCHAR2(50)	Yes	-	-
MNAME	VARCHAR2(50)	Yes	-	-
LNAME	VARCHAR2(50)	Yes	-	-
GENDER	VARCHAR2(50)	Yes	-	-
BDATE	VARCHAR2(50)	Yes	-	-
POSITION	VARCHAR2(50)	Yes	-	-
D_ID	NUMBER	Yes	-	-
1 - 12				



- Table DEPARTMENT:

```
CREATE TABLE DEPARTMENT (
DEPT_ID NUMBER NOT NULL PRIMARY KEY,
NAME VARCHAR2(50),
DEPT_MNGR VARCHAR2(50),
LOCATION VARCHAR2(255),
MNGR_SSN NUMBER NOT NULL,
FOREIGN KEY (MNGR_SSN) REFERENCES Employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
DEPT_ID	NUMBER	No	-	1
NAME	VARCHAR2(50)	Yes	-	-
DEPT_MNGR	VARCHAR2(50)	Yes	-	-
LOCATION	VARCHAR2(255)	Yes	-	-
MNGR_SSN	NUMBER	No	-	-
1 - 5				

- Table APPLICANT:

```
CREATE TABLE APPLICANT (
APP_ID NUMBER NOT NULL PRIMARY KEY,
NAME VARCHAR2(50),
BIO VARCHAR2(250),
EMAIL VARCHAR2(50),
PHONE_NUMBER VARCHAR2(50)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
APP_ID	NUMBER	No	-	1
NAME	VARCHAR2(50)	Yes	-	-
BIO	VARCHAR2(250)	Yes	-	-
EMAIL	VARCHAR2(50)	Yes	-	-
PHONE_NUMBER	VARCHAR2(50)	Yes	-	-
1 - 5				

- Table ATTENDANCE:

```
CREATE TABLE ATTENDANCE (
ATT_ID NUMBER NOT NULL PRIMARY KEY,
ATT_STATUS VARCHAR2(50),
ATT_DATE DATE,
TIME_IN TIMESTAMP (6),
TIME_OUT TIMESTAMP (6),
ESSN NUMBER NOT NULL,
FOREIGN KEY (ESSN) REFERENCES Employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
ATT_ID	NUMBER	No	-	1
ATT_STATUS	VARCHAR2(50)	Yes	-	-
ATT_DATE	DATE	Yes	-	-
TIME_IN	TIMESTAMP(6)	Yes	-	-
TIME_OUT	TIMESTAMP(6)	Yes	-	-
ESSN	NUMBER	No	-	-
1 - 6				

- Table BENEFITS:

```
CREATE TABLE BENEFITS (
BEN_ID NUMBER NOT NULL PRIMARY KEY,
COVERAGE_TYPE VARCHAR2(50),
COVERAGE_LEVEL VARCHAR2(50),
PERIOD VARCHAR2(50),
ESSN NUMBER NOT NULL,
FOREIGN KEY (ESSN) REFERENCES Employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
BEN_ID	NUMBER	No	-	1
COVERAGE_TYPE	VARCHAR2(50)	Yes	-	-
COVERAGE_LEVEL	VARCHAR2(50)	Yes	-	-
PERIOD	VARCHAR2(50)	Yes	-	-
ESSN	NUMBER	No	-	-
1 - 5				

▪ Table EVALUATION:

```
CREATE TABLE EVALUATION (
EVAL_ID NUMBER NOT NULL PRIMARY KEY,
EVALUATOR_ID NUMBER NOT NULL,
RATING NUMBER,
EVAL_DATE DATE,
FEEDBACK VARCHAR2(500),
ESSN NUMBER NOT NULL,
FOREIGN KEY (ESSN) REFERENCES employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
EVAL_ID	NUMBER	No	-	1
EVALUATOR_ID	NUMBER	No	-	-
RATING	NUMBER	Yes	-	-
EVAL_DATE	DATE	Yes	-	-
FEEDBACK	VARCHAR2(500)	Yes	-	-
ESSN	NUMBER	No	-	-
1 - 6				

▪ Table PAYROLL:

```
CREATE TABLE PAYROLL (
PAYROLL_ID NUMBER NOT NULL PRIMARY KEY,
CURRENCY VARCHAR2(50),
FREQUENCY VARCHAR2(250),
DEDUCTION NUMBER,
BONUS NUMBER,
SALARY NUMBER,
ESSN NUMBER NOT NULL,
FOREIGN KEY (ESSN) REFERENCES employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
PAYROLL_ID	NUMBER	No	-	1
CURRENCY	VARCHAR2(50)	Yes	-	-
FREQUENCY	VARCHAR2(250)	Yes	-	-
DEDUCTION	NUMBER	Yes	-	-
BONUS	NUMBER	Yes	-	-
SALARY	NUMBER	Yes	-	-
ESSN	NUMBER	No	-	-
1 - 7				

▪ Table DEPENDENT:

```
CREATE TABLE DEPENDENT (
ESSN NUMBER NOT NULL,
NAME VARCHAR2(50) NOT NULL,
GENDER VARCHAR2(50),
BIRTH_DATE DATE,
RELATIONSHIP VARCHAR2(50),
EDUCATION_LEVEL VARCHAR2(100),
PRIMARY KEY (ESSN, NAME)
FOREIGN KEY (ESSN) REFERENCES Employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
ESSN	NUMBER	No	-	1
NAME	VARCHAR2(50)	No	-	2
GENDER	VARCHAR2(50)	Yes	-	-
BIRTH_DATE	DATE	Yes	-	-
RELATIONSHIP	VARCHAR2(50)	Yes	-	-
EDUCATION_LEVEL	VARCHAR2(100)	Yes	-	-
1 - 6				

▪ Table RECRUITMENT:

```
CREATE TABLE RECRUITMENT (
REC_ID NUMBER NOT NULL PRIMARY KEY,
POSITION VARCHAR2(50),
DESCRIPTION VARCHAR2(250),
STATUS VARCHAR2(50),
TYPE VARCHAR2(50),
D_ID NUMBER NOT NULL,
FOREIGN KEY (D_ID) REFERENCES department (DEPT_ID));
```

Column Name	Data Type	Nullable	Default	Primary Key
REC_ID	NUMBER	No	-	1
POSITION	VARCHAR2(50)	Yes	-	-
DESCRIPTION	VARCHAR2(250)	Yes	-	-
STATUS	VARCHAR2(50)	Yes	-	-
TYPE	VARCHAR2(50)	Yes	-	-
D_ID	NUMBER	No	-	-
1 - 6				

▪ Table VACATION:

```
CREATE TABLE VACATION (
VAC_ID NUMBER NOT NULL PRIMARY KEY,
TYPE VARCHAR2(50),
START_DATE DATE,
END_DATE DATE,
DECISION VARCHAR2(50),
DECISION_BY NUMBER NOT NULL,
ESSN NUMBER NOT NULL,
FOREIGN KEY (ESSN) REFERENCES employee (SSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
VAC_ID	NUMBER	No	-	1
TYPE	VARCHAR2(50)	Yes	-	-
START_DATE	DATE	Yes	-	-
END_DATE	DATE	Yes	-	-
DECISION	VARCHAR2(50)	Yes	-	-
DECISION_BY	NUMBER	No	-	-
ESSN	NUMBER	No	-	-
1 - 7				

▪ Table APPLIES:

```
CREATE TABLE APPLIES (
APPL_ID NUMBER NOT NULL,
RECR_ID NUMBER NOT NULL,
APPLY_DATE DATE,
PRIMARY KEY (APPL_ID, RECR_ID),
FOREIGN KEY (APPL_ID) REFERENCES Applicant (APPL_ID),
FOREIGN KEY (RECR_ID) REFERENCES Recruitment (RECR_ID));
```

Column Name	Data Type	Nullable	Default	Primary Key
APPL_ID	NUMBER	No	-	1
RECR_ID	NUMBER	No	-	2
APPLY_DATE	DATE	Yes	-	-
1 - 3				

- Table ASSESSES:

```
CREATE TABLE ASSESSES (
APPL_ID NUMBER NOT NULL,
ESSN NUMBER NOT NULL,
PRIMARY KEY (APPL_ID, ESSN),
FOREIGN KEY (APPL_ID) REFERENCES Applicant (APPL_ID),
FOREIGN KEY (ESSN) REFERENCES Employee (ESSN)
);
```

Column Name	Data Type	Nullable	Default	Primary Key
APPL_ID	NUMBER	No	-	1
ESSN	NUMBER	No	-	2
				1 - 2

- Table PUBLISHES:

```
CREATE TABLE PUBLISHES (
ESSN NUMBER NOT NULL,
RECR_ID NUMBER NOT NULL,
PRIMARY KEY (ESSN, RECR_ID),
FOREIGN KEY (ESSN) REFERENCES employee (SSN),
FOREIGN KEY (RECR_ID) REFERENCES recruitment (REC_ID));
```

Column Name	Data Type	Nullable	Default	Primary Key
ESSN	NUMBER	No	-	1
RECR_ID	NUMBER	No	-	2
				1 - 2

- Table SKILLS:

```
CREATE TABLE skills (
SKILLS VARCHAR2(50) NOT NULL,
ESSN NUMBER NOT NULL,
PRIMARY KEY (SKILLS, ESSN),
FOREIGN KEY (ESSN) REFERENCES employee (SSN));
```

Column Name	Data Type	Nullable	Default	Primary Key
SKILLS	VARCHAR2(50)	No	-	1
ESSN	NUMBER	No	-	2
				1 - 2

- Table PROVIDER:

```
CREATE TABLE PROVIDER (
PROVIDER VARCHAR2(50) NOT NULL,
BENF_ID NUMBER NOT NULL,
PRIMARY KEY (PROVIDER, BENF_ID),
FOREIGN KEY (BENF_ID) REFERENCES benefits (BENF_ID));
```

Column Name	Data Type	Nullable	Default	Primary Key
PROVIDER	VARCHAR2(50)	No	-	1
BENF_ID	NUMBER	No	-	2
				1 - 2

## IX. Insertion of Data:

The INSERT INTO query is used to add new records into an existing table within the database. It specifies the table name and the columns into which data will be inserted, followed by the values to be inserted into each corresponding column. The query begins with the INSERT INTO statement followed by the name of the table where data will be inserted. Inside the parentheses, the column names of the table are listed to indicate where the values will be inserted. Following the column names, the VALUES keyword is used to specify the actual values to be inserted into each column. The values are provided within parentheses and separated by commas, following the same order as the column names listed earlier. Overall, the INSERT INTO query adds new records into a table by specifying the table name, the columns where data will be inserted, and the corresponding values for each column.

▪ Table EMPLOYEE:

INSERT INTO EMPLOYEE VALUES (1, 'emp1@gmail.com', '12345678', 'full time', 'aley', 'john', 'smith', 'doe', 'male', '2000-02-02', 'engineer', 1);

INSERT INTO EMPLOYEE VALUES (15, 'emp15@gmail.com', '8648494', 'full time', 'jbeil', 'youssef', 'youssef', 'borji', 'male', '1993-05-29', 'finance', 5);

INSERT INTO EMPLOYEE VALUES (17, 'emp17@gmail.com', '23989184', 'part time', 'dbayye', 'sara', 'mohammad', 'el-halabi', 'female', '1976-12-12', 'services specialist', 7);

INSERT INTO EMPLOYEE VALUES (18, 'emp18@gmail.co', '2348234', 'full time', 'rashya', 'khaled', 'ali', 'Ahmad', 'male', '2000-09-18', 'marketing', 8);

INSERT INTO EMPLOYEE VALUES (19, 'emp19@gmial.com', '82349782', 'full time', 'zahle', 'christine', 'ali', 'Ahmad', 'female', '1991-02-19', 'marketing', 8);

INSERT INTO EMPLOYEE VALUES (20, 'emp20@gmail.com', '72364763', 'full time', 'batroun', 'aleen', 'mostapha', 'jaber', 'female', '1998-08-21', 'sales', 9);

INSERT INTO EMPLOYEE VALUES (21, 'emp21@mail.com', '83468234', 'full time', 'beirut', 'amine', 'naim', 'Abou chkra', 'male', '1999-09-09', 'lawyer', 10);

INSERT INTO EMPLOYEE VALUES (14, 'emp14@gmail.com', '12348765', 'full time', 'aley', 'louva', 'cupper', 'doe', 'female', '2004-09-28', 'HR Supervisor', 3);

INSERT INTO EMPLOYEE VALUES (2, 'emp2@gmail.com', '22345678', 'full time', 'Beirut', 'Jad', 'Jamil', 'Al-Awar', 'male', '2004-09-28', 'ceo', 1);

INSERT INTO EMPLOYEE VALUES (4, 'emp4@gmail.com', '7654321', 'part time', 'Tripoli', 'George', 'Najm', 'Sabbagh', 'male', '1990-05-15', 'developer', 2);

INSERT INTO EMPLOYEE VALUES (5, 'emp5@gmail.com', '11223344', 'full time', 'Jounieh', 'Sara', 'Fawaz', 'Dagher', 'female', '1985-07-20', 'IT consultant', 6);

INSERT INTO EMPLOYEE VALUES (6, 'emp6@gmail.com', '99988877', 'part time', 'Zahle', 'Nabil', 'Khoury', 'Haddad', 'male', '1978-12-10', 'accountant', 4);

INSERT INTO EMPLOYEE VALUES (7, 'emp7@gmail.com', '6543210', 'full time', 'Sidon', 'Lara', 'Sleiman', 'Hajj', 'female', '1993-03-25', 'engineer', 1);

INSERT INTO EMPLOYEE VALUES (8, 'emp8@gmail.com', '4567890', 'part time', 'Baabda', 'Tony', 'Moussa', 'Antoun', 'male', '1982-09-08', 'designer', 2);

INSERT INTO EMPLOYEE VALUES (9, 'emp9@gmail.com', '9876543', 'full time', 'Aley', 'Rita', 'Haddad', 'Nasser', 'female', '1987-11-17', 'representative', 7);

INSERT INTO EMPLOYEE VALUES (10, 'emp10@gmail.com', '33445566', 'part time', 'Zgharta', 'Khaled', 'Salameh', 'Matar', 'male', '1995-06-30', 'analyst', 4);

INSERT INTO EMPLOYEE VALUES (11, 'emp11@gmail.com', '88997766', 'full time', 'Nabatieh', 'Nour', 'Assaf', 'Hajj', 'female', '1991-08-12', 'coordinator', 6);

INSERT INTO EMPLOYEE VALUES (12, 'emp12@gmail.com', '11223344', 'part time', 'Batroun', 'Elie', 'Ziadeh', 'Asmar', 'male', '1980-04-05', 'HR assistant', 3);

INSERT INTO EMPLOYEE VALUES (13, 'emp13@gmail.com', '99887766', 'full time', 'Tyre', 'Maya', 'Fadel', 'Khair', 'female', '1984-10-22', 'HR specialist', 3);

INSERT INTO EMPLOYEE VALUES (3, 'emp3@gmail.com', '11990077', 'full time', 'daheyyi', 'mostapha', 'mostapha', 'zein-eddine', 'male', '1999-01-01', 'IT assistant', 6);

INSERT INTO EMPLOYEE VALUES (16, 'emp16@gmail..com', '98468476', 'part time', 'zahle', 'lina', 'ahmad', 'el-halabi', 'female', '1987-09-16', 'analyst', 5);



EDIT	SSN	EMAIL	PH_NUM	STATUS	ADDRESS	FNAME	MNAME	LNAME	GENDER	BDATE	POSITION	D_ID
	1	emp1@gmail.com	12345678	full time	aley	john	smith	doe	male	2000-02-02	engineer	1
	15	emp15@gmail.com	8648494	full time	jbeil	youssef	youssef	borji	male	1993-05-29	finance	5
	17	emp17@gmail.com	23989184	part time	dbayye	sara	mohammad	el-halabi	female	1976-12-12	services specialist	7
	18	emp18@gmail.co	2348234	full time	rashya	khaled	ali	Ahmad	male	2000-09-18	marketing	8
	19	emp19@gmail.com	82349782	full time	zahle	christine	ali	Ahmad	female	1991-02-19	marketing	8
	20	emp20@gmail.com	72364763	full time	batroun	aleen	mostapha	jaber	female	1998-08-21	sales	9
	21	emp21@gmail.com	83468234	full time	beirut	amine	naim	Abou chkra	male	1999-09-09	lawyer	10
	14	emp14@gmail.com	12348765	full time	aley	louva	cupper	doe	female	2004-09-28	HR Supervisor	3
	2	emp2@gmail.com	22345678	full time	Beirut	Jad	Jamil	Al-Awar	male	2004-09-28	ceo	1
	4	emp4@gmail.com	7654321	part time	Tripoli	George	Najm	Sabbagh	male	1990-05-15	developer	2
	5	emp5@gmail.com	11223344	full time	Jounieh	Sara	Fawaz	Dagher	female	1985-07-20	IT consultant	6
	6	emp6@gmail.com	99988877	part time	Zahle	Nabil	Khoury	Haddad	male	1978-12-10	accountant	4
	7	emp7@gmail.com	6543210	full time	Sidon	Lara	Sleiman	Hajj	female	1993-03-25	engineer	1
	8	emp8@gmail.com	4567890	part time	Baabda	Tony	Moussa	Antoun	male	1982-09-08	designer	2
	9	emp9@gmail.com	9876543	full time	Aley	Rita	Haddad	Nasser	female	1987-11-17	representative	7

EDIT	SSN	EMAIL	PH_NUM	STATUS	ADDRESS	FNAME	MNAME	LNAME	GENDER	BDATE	POSITION	D_ID
	10	emp10@gmail.com	33445566	part time	Zgharta	Khaled	Salameh	Matar	male	1995-06-30	analyst	4
	11	emp11@gmail.com	88997766	full time	Nabatieh	Nour	Assaf	Hajj	female	1991-08-12	coordinator	6
	12	emp12@gmail.com	11223344	part time	Batroun	Elie	Ziadeh	Asmar	male	1980-04-05	HR assistant	3
	13	emp13@gmail.com	99887766	full time	Tyre	Maya	Fadel	Khair	female	1984-10-22	HR specialist	3
	3	emp3@gmail.com	11990077	full time	daheyyi	mostapha	mostapha	zein-eddine	male	1999-01-01	IT assistant	6
	16	emp16@gmail.com	98468476	part time	zahle	lina	ahmad	el-halabi	female	1987-09-16	analyst	5

row(s) 16 - 21 of 21

■ Table DEPARTMENT:

INSERT INTO DEPARTMENT VALUES (10, 'legal', 'amine abou chkra', 'beirut', 21);

INSERT INTO DEPARTMENT VALUES (1, 'Engineering', 'john doe', 'beirut', 2);

INSERT INTO DEPARTMENT VALUES (2, 'Development', 'George sabbagh', 'beirut', 4);

INSERT INTO DEPARTMENT VALUES (9, 'sales', 'aleen jaber', 'beirut', 20);

INSERT INTO DEPARTMENT VALUES (3, 'Human Resources', 'louva smith', 'beirut', 14);











INSERT INTO DEPARTMENT VALUES (4, 'accounting', 'nabil haddad', 'beirut', 6);

INSERT INTO DEPARTMENT VALUES (5, 'finance', 'yousef borji', 'beirut', 15);

INSERT INTO DEPARTMENT VALUES (6, 'IT', 'Sara dagher', 'beirut', 5);

INSERT INTO DEPARTMENT VALUES (7, 'customer services', 'rita nasser', 'beirut', 9);

INSERT INTO DEPARTMENT VALUES (8, 'marketing', 'khaled ahmad', 'beirut', 18);

EDIT	DEPT_ID	NAME	DEPT_MNGR	LOCATION	MNGR_SSN
	10	legal	amine abou chkra	beirut	21
	1	Engineering	john doe	beirut	2
	2	Development	George sabbagh	beirut	4
	9	sales	aleen jaber	beirut	20
	3	Human Resources	louva smith	beirut	14
	4	accounting	nabil haddad	beirut	6
	5	finance	yousef borji	beirut	15
	6	IT	Sara dagher	beirut	5
	7	customer services	rita nasser	beirut	9
	8	marketing	khaled ahmad	beirut	18
row(s) 1 - 10 of 10					

▪ Table APPLICANT:

INSERT INTO APPLICANT VALUES (1, 'Elias Nader', 'Experienced software engineer with 5+ years of experience in web development.', 'app1@example.com', '7328762');

INSERT INTO APPLICANT VALUES (2, 'Layla Fakhoury', 'Recent graduate with a degree in computer science, eager to start a career in data analysis.', 'app2@example.com', '12234567');

INSERT INTO APPLICANT VALUES (3, 'Karim Nassar', 'Skilled project manager with a background in leading cross-functional teams.', 'app3@example.com', '89345678');

INSERT INTO APPLICANT VALUES (4, 'Rima Haddad', 'Creative graphic designer specializing in branding and visual identity.', 'app4@example.com', '55456789');

INSERT INTO APPLICANT VALUES (5, 'Nour Khalil', 'Experienced sales professional with a track record of exceeding targets.', 'app5@example.com', '40567890');

INSERT INTO APPLICANT VALUES (6, 'Rana Abi-Nader', 'Motivated marketing specialist with expertise in digital marketing strategies.', 'app6@example.com', '30678901');

INSERT INTO APPLICANT VALUES (7, 'Omar Saliba', 'Detail-oriented accountant with experience in financial analysis and reporting.', 'app7@example.com', '91789012');

INSERT INTO APPLICANT VALUES (8, 'Jana Haddad', 'Results-driven HR manager with experience in recruitment and employee relations.', 'app8@example.com', '22890123');

INSERT INTO APPLICANT VALUES (9, 'Rami Harb', 'Dynamic product manager with a passion for creating innovative solutions.', 'app9@example.com', '83901234');

INSERT INTO APPLICANT VALUES (10, 'Lea Abi-Rizk', 'Dedicated customer service representative committed to providing exceptional support.', 'app10@example.com', '42012345');

EDIT	APP_ID	NAME	BIO	EMAIL	PHONE_NUMBER
	1	Elias Nader	Experienced software engineer with 5+ years of experience in web development.	app1@example.com	7328762
	2	Layla Fakhoury	Recent graduate with a degree in computer science, eager to start a career in data analysis.	app2@example.com	12234567
	3	Karim Nassar	Skilled project manager with a background in leading cross-functional teams.	app3@example.com	89345678
	4	Rima Haddad	Creative graphic designer specializing in branding and visual identity.	app4@example.com	55456789
	5	Nour Khalil	Experienced sales professional with a track record of exceeding targets.	app5@example.com	40567890
	6	Rana Abi-Nader	Motivated marketing specialist with expertise in digital marketing strategies.	app6@example.com	30678901
	7	Omar Saliba	Detail-oriented accountant with experience in financial analysis and reporting.	app7@example.com	91789012
	8	Jana Haddad	Results-driven HR manager with experience in recruitment and employee relations.	app8@example.com	22890123
	9	Rami Harb	Dynamic product manager with a passion for creating innovative solutions.	app9@example.com	83901234
	10	Lea Abi-Rizk	Dedicated customer service representative committed to providing exceptional support.	app10@example.com	42012345
row(s) 1 - 10 of 10					

▪ Table ATTENDANCE:

INSERT INTO ATTENDANCE VALUES (1, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 1);

INSERT INTO ATTENDANCE VALUES (2, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 08.59.01.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.01.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 2);

INSERT INTO ATTENDANCE VALUES (3, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 08.59.01.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.01.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 3);

INSERT INTO ATTENDANCE VALUES (4, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 08.59.01.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.01.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 4);

INSERT INTO ATTENDANCE VALUES (7, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.10.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 7);

INSERT INTO ATTENDANCE VALUES (8, 'absent', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), NULL, NULL, 8);

INSERT INTO ATTENDANCE VALUES (15, 'absent', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), NULL, NULL, 15);

INSERT INTO ATTENDANCE VALUES (9, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 9);

INSERT INTO ATTENDANCE VALUES (10, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 10);

INSERT INTO ATTENDANCE VALUES (11, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 11);

INSERT INTO ATTENDANCE VALUES (12, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 12);

INSERT INTO ATTENDANCE VALUES (16, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 16);

INSERT INTO ATTENDANCE VALUES (17, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.30.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 17);

INSERT INTO ATTENDANCE VALUES (21, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 21);

INSERT INTO ATTENDANCE VALUES (6, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 06.01.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 6);

















INSERT INTO ATTENDANCE VALUES (13, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 13);

INSERT INTO ATTENDANCE VALUES (14, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 14);

INSERT INTO ATTENDANCE VALUES (18, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 18);

INSERT INTO ATTENDANCE VALUES (19, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 19);

INSERT INTO ATTENDANCE VALUES (20, 'Present', TO\_DATE('04/02/2024', 'MM/DD/YYYY'),  
TO\_TIMESTAMP('02-APR-24 09.00.00.000000 AM', 'DD-MON-RR HH.MI.SS.FF AM'), TO\_TIMESTAMP('02-  
APR-24 05.00.00.000000 PM', 'DD-MON-RR HH.MI.SS.FF PM'), 20);

EDIT	ATT_ID	ATT_STATUS	ATT_DATE	TIME_IN	TIME_OUT	ESSN
	1	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	1
	2	Present	04/02/2024	02-APR-24 08.59.01.000000 AM	02-APR-24 05.01.00.000000 PM	2
	3	Present	04/02/2024	02-APR-24 08.59.01.000000 AM	02-APR-24 05.01.00.000000 PM	3
	4	Present	04/02/2024	02-APR-24 08.59.01.000000 AM	02-APR-24 05.01.00.000000 PM	4
	7	Present	04/02/2024	02-APR-24 09.10.00.000000 AM	02-APR-24 05.00.00.000000 PM	7
	8	absent	04/02/2024	-	-	8
	15	absent	04/02/2024	-	-	15
	9	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	9
	10	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	10
	11	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	11
	12	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	12
	16	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	16
	17	Present	04/02/2024	02-APR-24 09.30.00.000000 AM	02-APR-24 05.00.00.000000 PM	17
	21	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	21
	5	Present	04/02/2024	02-APR-24 09.05.00.000000 AM	02-APR-24 05.01.00.000000 PM	5
						row(s) 1 - 15 of 21 

EDIT	ATT_ID	ATT_STATUS	ATT_DATE	TIME_IN	TIME_OUT	ESSN
	6	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 06.01.00.000000 PM	6
	13	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	13
	14	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	14
	18	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	18
	19	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	19
	20	Present	04/02/2024	02-APR-24 09.00.00.000000 AM	02-APR-24 05.00.00.000000 PM	20
row(s) 16 - 21 of 21						

▪ Table BENEFITS:

```

INSERT INTO BENEFITS VALUES (1, 'health insurance', 'A', '5 years', 1);
INSERT INTO BENEFITS VALUES (2, 'health insurance', 'B', '5 years', 2);
INSERT INTO BENEFITS VALUES (3, 'life insurance', 'A', '10 years', 2);
INSERT INTO BENEFITS VALUES (4, 'health insurance', 'A', '5 years', 3);
INSERT INTO BENEFITS VALUES (7, 'disability insurance', 'A', '10 years', 5);
INSERT INTO BENEFITS VALUES (8, 'health insurance', 'A', '5 years', 6);
INSERT INTO BENEFITS VALUES (11, 'health insurance', 'A', '5 years', 9);
INSERT INTO BENEFITS VALUES (13, 'health insurance', 'A', '5 years', 11);
INSERT INTO BENEFITS VALUES (14, 'health insurance', 'A', '5 years', 12);
INSERT INTO BENEFITS VALUES (15, 'health insurance', 'A', '5 years', 13);
INSERT INTO BENEFITS VALUES (16, 'health insurance', 'A', '5 years', 14);
INSERT INTO BENEFITS VALUES (18, 'health insurance', 'A', '5 years', 15);
INSERT INTO BENEFITS VALUES (19, 'health insurance', 'A', '5 years', 16);
INSERT INTO BENEFITS VALUES (20, 'health insurance', 'A', '5 years', 17);
INSERT INTO BENEFITS VALUES (21, 'health insurance', 'A', '5 years', 18);
INSERT INTO BENEFITS VALUES (23, 'health insurance', 'A', '5 years', 20);
INSERT INTO BENEFITS VALUES (24, 'health insurance', 'A', '5 years', 21);
INSERT INTO BENEFITS VALUES (25, 'health insurance', 'A', '5 years', 4);
INSERT INTO BENEFITS VALUES (5, 'retirement plan', '-', 'forever', 1);
INSERT INTO BENEFITS VALUES (6, 'retirement plan', '-', 'forever', 2);
INSERT INTO BENEFITS VALUES (9, 'health insurance', 'A', '5 years', 7);
INSERT INTO BENEFITS VALUES (10, 'health insurance', 'A', '5 years', 8);
INSERT INTO BENEFITS VALUES (12, 'health insurance', 'A', '5 years', 10);
INSERT INTO BENEFITS VALUES (22, 'health insurance', 'A', '5 years', 19);
INSERT INTO BENEFITS VALUES (26, 'health insurance', 'A', '5 years', 5);

```

INSERT INTO BENEFITS VALUES (27, 'health insurance', 'A', '5 years', 7);

INSERT INTO BENEFITS VALUES (28, 'life insurance', 'A', '5 years', 19);

EDIT	BEN_ID	COVERAGE_TYPE	COVERAGE_LEVEL	PERIOD	ESSN
	1	health insurance	A	5 years	1
	2	health insurance	B	5 years	2
	3	life insurance	A	10 years	2
	4	health insurance	A	5 years	3
	7	disability insurance	A	10 years	5
	8	health insurance	A	5 years	6
	11	health insurance	A	5 years	9
	13	health insurance	A	5 years	11
	14	health insurance	A	5 years	12
	15	health insurance	A	5 years	13
	16	health insurance	A	5 years	14
	18	health insurance	A	5 years	15
	19	health insurance	A	5 years	16
	20	health insurance	A	5 years	17
	21	health insurance	A	5 years	18
					row(s) 1 - 15 of 27

EDIT	BEN_ID	COVERAGE_TYPE	COVERAGE_LEVEL	PERIOD	ESSN
	23	health insurance	A	5 years	20
	24	health insurance	A	5 years	21
	25	health insurance	A	5 years	4
	5	retirement plan	-	forever	1
	6	retirement plan	-	forever	2
	9	health insurance	A	5 years	7
	10	health insurance	A	5 years	8
	12	health insurance	A	5 years	10
	22	health insurance	A	5 years	19
	26	health insurance	A	5 years	5
	27	health insurance	A	5 years	7
	28	life insurance	A	5 years	19
					row(s) 16 - 27 of 27

■ Table EVALUATION:

INSERT INTO EVALUATION VALUES (1, 2, 8, '2024-03-15', 'Excellent performance overall.', 1);

INSERT INTO EVALUATION VALUES (2, 2, 9, '2024-04-02', 'Outstanding work, exceeded expectations.', 21);

INSERT INTO EVALUATION VALUES (3, 2, 7, '2024-04-02', 'Good effort, room for improvement in certain areas.', 20);

INSERT INTO EVALUATION VALUES (4, 2, 5, '2024-04-02', 'Below expectations, needs improvement.', 5);

INSERT INTO EVALUATION VALUES (10, 15, 4, '2024-04-02', 'Performance below expectations, improvement needed in multiple areas.', 16);











INSERT INTO EVALUATION VALUES (5, 2, 8, '2024-04-02', 'Consistently good performance.', 18);

INSERT INTO EVALUATION VALUES (6, 4, 8, '2024-04-02', 'Solid performance, with areas for growth.', 10);

INSERT INTO EVALUATION VALUES (7, 14, 10, '2024-04-02', 'Exceptional performance, top-notch work.', 12);

INSERT INTO EVALUATION VALUES (8, 14, 8, '2024-04-02', 'Consistently outstanding, a model employee.', 13);

INSERT INTO EVALUATION VALUES (9, 18, 7, '2024-04-02', 'Consistently outstanding, a model employee.', 19);

EDIT	EVAL_ID	EVALUATOR_ID	RATING	EVAL_DATE	FEEDBACK	ESSN
	1	2	8	03/15/2024	Excellent performance overall.	1
	2	2	9	04/02/2024	Outstanding work, exceeded expectations.	21
	3	2	7	04/02/2024	Good effort, room for improvement in certain areas.	20
	4	2	5	04/02/2024	Below expectations, needs improvement.	5
	10	15	4	04/02/2024	Performance below expectations, improvement needed in multiple areas.	16
	5	2	8	04/02/2024	Consistently good performance.	18
	6	4	8	04/02/2024	Solid performance, with areas for growth.	10
	7	14	10	04/02/2024	Exceptional performance, top-notch work.	12
	8	14	8	04/02/2024	Consistently outstanding, a model employee.	13
	9	18	7	04/02/2024	Consistently outstanding, a model employee.	19
row(s) 1 - 10 of 10						

▪ Table PAYROLL:

```

INSERT INTO PAYROLL VALUES (3, 'USD', 'Monthly', 150, 100, 9000, 4);
INSERT INTO PAYROLL VALUES (4, 'USD', 'Monthly', 150, 200, 8700, 20);
INSERT INTO PAYROLL VALUES (5, 'USD', 'Monthly', 0, 200, 10500, 14);
INSERT INTO PAYROLL VALUES (6, 'USD', 'Monthly', 75, 200, 9200, 6);
INSERT INTO PAYROLL VALUES (9, 'USD', 'Monthly', 305, 410, 9900, 9);
INSERT INTO PAYROLL VALUES (10, 'USD', 'Monthly', 175, 400, 10000, 18);
INSERT INTO PAYROLL VALUES (13, 'USD', 'Monthly', 175, 100, 6000, 7);
INSERT INTO PAYROLL VALUES (14, 'USD', 'Monthly', 75, 100, 4000, 8);
INSERT INTO PAYROLL VALUES (15, 'USD', 'Monthly', 300, 1000, 11000, 10);
INSERT INTO PAYROLL VALUES (16, 'USD', 'Monthly', 300, 100, 5000, 11);
INSERT INTO PAYROLL VALUES (17, 'USD', 'Monthly', 300, 255, 5000, 12);
INSERT INTO PAYROLL VALUES (18, 'USD', 'Monthly', 140, 255, 5000, 13);
INSERT INTO PAYROLL VALUES (20, 'USD', 'Monthly', 140, 225, 5000, 17);
INSERT INTO PAYROLL VALUES (21, 'USD', 'Monthly', 160, 213, 5000, 19);
INSERT INTO PAYROLL VALUES (1, 'USD', 'Monthly', 100, 200, 10660, 21);

```



EDIT	PAYROLL_ID	CURRENCY	FREQUENCY	DEDUCTION	BONUS	SALARY	ESSN
	3	USD	Monthly	150	100	9000	4
	4	USD	Monthly	150	200	8700	20
	5	USD	Monthly	0	200	10500	14
	6	USD	Monthly	75	200	9200	6
	9	USD	Monthly	305	410	9900	9
	10	USD	Monthly	175	400	10000	18
	13	USD	Monthly	175	100	6000	7
	14	USD	Monthly	75	100	4000	8
	15	USD	Monthly	300	1000	11000	10
	16	USD	Monthly	300	100	5000	11
	17	USD	Monthly	300	255	5000	12
	18	USD	Monthly	140	255	5000	13
	20	USD	Monthly	140	225	5000	17
	21	USD	Monthly	160	213	5000	19
	1	USD	Monthly	100	200	10660	21

EDIT	PAYROLL_ID	CURRENCY	FREQUENCY	DEDUCTION	BONUS	SALARY	ESSN
	2	USD	Monthly	50	600	25000	2
	7	USD	Monthly	75	200	10000	15
	8	USD	Monthly	305	200	10000	5
	11	USD	Monthly	175	400	5000	1
	12	USD	Monthly	175	200	5000	3
	19	USD	Monthly	140	235	5000	16
















⏪ row(s) 16 - 21 of 21

▪ Table DEPENDENT:

INSERT INTO DEPENDENT VALUES (1, 'jana', 'female', '2009-05-04', 'daughter', 'grade 9');  
 INSERT INTO DEPENDENT VALUES (1, 'ruba', 'female', '2000-02-02', 'wife', 'bachelors in marketing');  
 INSERT INTO DEPENDENT VALUES (3, 'fatima', 'female', '1999-02-03', 'wife', 'PHD in mathematics');  
 INSERT INTO DEPENDENT VALUES (4, 'malak', 'female', '1978-09-28', 'wife', NULL);  
 INSERT INTO DEPENDENT VALUES (4, 'jad', 'male', '2009-05-09', 'son', 'grade 9');  
 INSERT INTO DEPENDENT VALUES (4, 'maria', 'female', '2010-09-27', 'daughter', 'grade 8');  
 INSERT INTO DEPENDENT VALUES (5, 'ali', 'male', '1980-09-09', 'husband', 'engineer');  
 INSERT INTO DEPENDENT VALUES (5, 'siham', 'female', '2005-08-07', 'daughter', 'first year university');  
 INSERT INTO DEPENDENT VALUES (5, 'alia', 'female', '2006-04-22', 'daughter', 'grade 12');



INSERT INTO DEPENDENT VALUES (5, 'zainab', 'female', '2007-05-18', 'daughter', 'grade 11');  
 INSERT INTO DEPENDENT VALUES (5, 'julia', 'female', '2008-11-11', 'daughter', 'grade 10');  
 INSERT INTO DEPENDENT VALUES (5, 'hasan', 'male', '2010-06-06', 'son', 'grade 8');  
 INSERT INTO DEPENDENT VALUES (14, 'hero', 'male', '2004-04-28', 'husband', 'bachelor in computer science');  
 INSERT INTO DEPENDENT VALUES (14, 'max', 'male', '2023-01-11', 'son', '-');  
 INSERT INTO DEPENDENT VALUES (14, 'lucy', 'female', '2024-04-02', 'daughter', '-');

EDIT	ESSN	NAME	GENDER	BIRTH_DATE	RELATIONSHIP	EDUCATION_LEVEL
	1	jana	female	05/04/2009	daughter	grade 9
	1	ruba	female	02/02/2000	wife	bachelors in marketing
	3	fatima	female	02/03/1999	wife	PHD in mathematics
	4	malak	female	09/28/1978	wife	null
	4	jad	male	05/09/2009	son	grade 9
	4	maria	female	09/27/2010	daughter	grade 8
	5	ali	male	09/09/1980	husband	engineer
	5	siham	female	08/07/2005	daughter	first year university
	5	alia	female	04/22/2006	daughter	grade 12
	5	zainab	female	05/18/2007	daughter	grade 11
	5	julia	female	11/11/2008	daughter	grade 10
	5	hasan	male	06/06/2010	son	grade 8
	14	hero	male	04/28/2004	husband	bachelor in computer science
	14	max	male	01/11/2023	son	-
	14	lucy	female	04/02/2024	daughter	-
row(s) 1 - 15 of 15						

▪ Table RECRUITMENT:

INSERT INTO RECRUITEMENT VALUES (1, 'Software Engineer', 'Seeking an experienced software engineer to join our web development team.', 'Open', 'Full-time', 2),

INSERT INTO RECRUITEMENT VALUES (2, 'Data Analyst', 'Looking for a recent graduate in computer science to analyze data and generate insights.', 'Open', 'internship', 8),

INSERT INTO RECRUITEMENT VALUES (9, 'Product Manager', 'Searching for a dynamic product manager to innovate and drive product development.', 'Open', 'Full-time', 9),

INSERT INTO RECRUITEMENT VALUES (10, 'Customer Service Representative', 'In need of a dedicated customer service representative to provide exceptional support.', 'Open', 'part-time', 7),

INSERT INTO RECRUITEMENT VALUES (3, 'Project Manager', 'Hiring a skilled project manager to lead cross-functional teams and oversee project delivery.', 'Open', 'part-time', 5),

INSERT INTO RECRUITEMENT VALUES (4, 'Graphic Designer', 'Searching for a creative graphic designer to develop branding and visual identity materials.', 'Open', 'part-time', 4),

INSERT INTO RECRUITEMENT VALUES (6, 'Marketing Specialist', 'Seeking a motivated marketing specialist to develop and execute digital marketing strategies.', 'Open', 'Full-time', 8),

INSERT INTO RECRUITEMENT VALUES (7, 'Accountant', 'Looking for a detail-oriented accountant to handle financial analysis and reporting tasks.', 'Open', 'internship', 4),

INSERT INTO RECRUITEMENT VALUES (8, 'HR Manager', 'Hiring a results-driven HR manager to lead recruitment and employee relations efforts.', 'Open', 'Full-time', 3),

INSERT INTO RECRUITEMENT VALUES (5, 'Sales Representative', 'In need of an experienced sales professional to drive sales and meet targets.', 'Open', 'Full-time', 5);

EDIT	REC_ID	POSITION	DESCRIPTION	STATUS	TYPE	D_ID
	1	Software Engineer	Seeking an experienced software engineer to join our web development team.	Open	Full-time	2
	2	Data Analyst	Looking for a recent graduate in computer science to analyze data and generate insights.	Open	internship	8
	9	Product Manager	Searching for a dynamic product manager to innovate and drive product development.	Open	Full-time	9
	10	Customer Service Representative	In need of a dedicated customer service representative to provide exceptional support.	Open	part-time	7
	3	Project Manager	Hiring a skilled project manager to lead cross-functional teams and oversee project delivery.	Open	part-time	5
	4	Graphic Designer	Searching for a creative graphic designer to develop branding and visual identity materials.	Open	part-time	4
	6	Marketing Specialist	Seeking a motivated marketing specialist to develop and execute digital marketing strategies.	Open	Full-time	8
	7	Accountant	Looking for a detail-oriented accountant to handle financial analysis and reporting tasks.	Open	internship	4
	8	HR Manager	Hiring a results-driven HR manager to lead recruitment and employee relations efforts.	Open	Full-time	3
	5	Sales Representative	In need of an experienced sales professional to drive sales and meet targets.	Open	Full-time	5
row(s) 1 - 10 of 10						

#### ■ Table VACATION:

INSERT INTO VACATION VALUES (4, 'Annual Leave', TO\_DATE('06/20/2024', 'MM/DD/YYYY'), TO\_DATE('07/05/2024', 'MM/DD/YYYY'), 'approved', 6, 10);

INSERT INTO VACATION VALUES (2, 'Sick Leave', TO\_DATE('05/10/2024', 'MM/DD/YYYY'), TO\_DATE('05/15/2024', 'MM/DD/YYYY'), 'Pending', 2, 6, 18);

INSERT INTO VACATION VALUES (3, 'Maternity Leave', TO\_DATE('06/20/2024', 'MM/DD/YYYY'), TO\_DATE('07/05/2024', 'MM/DD/YYYY'), 'approved', 5, 5);

INSERT INTO VACATION VALUES (5, 'Sick Leave', TO\_DATE('04/02/2024', 'MM/DD/YYYY'), TO\_DATE('07/02/2024', 'MM/DD/YYYY'), 'rejected', 2, 9);

INSERT INTO VACATION VALUES (6, 'Unpaid Leave', TO\_DATE('03/10/2024', 'MM/DD/YYYY'), TO\_DATE('03/15/2024', 'MM/DD/YYYY'), 'pending', 4, 8);

INSERT INTO VACATION VALUES (7, 'Unpaid Leave', TO\_DATE('03/10/2024', 'MM/DD/YYYY'), TO\_DATE('03/15/2024', 'MM/DD/YYYY'), 'pending', 15, 17);

INSERT INTO VACATION VALUES (1, 'Annual Leave', TO\_DATE('04/01/2024', 'MM/DD/YYYY'), TO\_DATE('04/05/2024', 'MM/DD/YYYY'), 'Approved', 2, 1);

INSERT INTO VACATION VALUES (8, 'Annual Leave', TO\_DATE('05/15/2024', 'MM/DD/YYYY'), TO\_DATE('06/15/2024', 'MM/DD/YYYY'), 'pending', 14, 12);

INSERT INTO VACATION VALUES (9, 'Annual Leave', TO\_DATE('05/15/2024', 'MM/DD/YYYY'), TO\_DATE('06/15/2024', 'MM/DD/YYYY'), 'approved', 17, 9);

INSERT INTO VACATION VALUES (10, 'Annual Leave', TO\_DATE('05/15/2024', 'MM/DD/YYYY'), TO\_DATE('06/15/2024', 'MM/DD/YYYY'), 'rejected', 19, 18);

EDIT	VAC_ID	TYPE	START_DATE	END_DATE	DECISION	DECISION_BY	ESSN
	4	Annual Leave	06/20/2024	07/05/2024	approved	6	10
	2	Sick Leave	05/10/2024	05/15/2024	Pending	2	18
	3	Maternity Leave	06/20/2024	07/05/2024	approved	5	5
	5	Sick Leave	04/02/2024	07/02/2024	rejected	2	9
	6	Unpaid Leave	03/10/2024	03/15/2024	pending	4	8
	7	Unpaid Leave	03/10/2024	03/15/2024	pending	15	17
	1	Annual Leave	04/01/2024	04/05/2024	Approved	2	1
	8	Annual Leave	05/15/2024	06/15/2024	pending	14	12
	9	Annual Leave	05/15/2024	06/15/2024	approved	17	9
	10	Annual Leave	05/15/2024	06/15/2024	rejected	19	18
							row(s) 1 - 10 of 10

▪ Table APPLIES:

INSERT INTO APPLIES VALUES (1, 1, TO\_DATE ('04/01/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (4, 4, TO\_DATE ('04/04/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (5, 5, TO\_DATE ('04/05/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (6, 6, TO\_DATE ('04/06/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (7, 7, TO\_DATE ('04/07/2024', 'MM/DD/YYYY'));











INSERT INTO APPLIES VALUES (10, 10, TO\_DATE ('04/10/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (2, 2, TO\_DATE ('04/02/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (3, 3, TO\_DATE ('04/03/2024', 'MM/DD/YYYY'));











INSERT INTO APPLIES VALUES (8, 8, TO\_DATE ('04/08/2024', 'MM/DD/YYYY'));

INSERT INTO APPLIES VALUES (9, 9, TO\_DATE ('04/09/2024', 'MM/DD/YYYY'));

EDIT	APPL_ID	RECR_ID	APPLY_DATE
	1	1	04/01/2024
	4	4	04/04/2024
	5	5	04/05/2024
	6	6	04/06/2024
	7	7	04/07/2024
	10	10	04/10/2024
	2	2	04/02/2024
	3	3	04/03/2024
	8	8	04/08/2024
	9	9	04/09/2024
row(s) 1 - 10 of 10			

▪ Table ASSESSES:

INSERT INTO ASSESSES VALUES (1, 12);  
 INSERT INTO ASSESSES VALUES (2, 13);  
 INSERT INTO ASSESSES VALUES (3, 14);  
 INSERT INTO ASSESSES VALUES (4, 12);  
 INSERT INTO ASSESSES VALUES (5, 13);  
 INSERT INTO ASSESSES VALUES (6, 14);  
 INSERT INTO ASSESSES VALUES (7, 12);  
 INSERT INTO ASSESSES VALUES (8, 13);  
 INSERT INTO ASSESSES VALUES (9, 14);  
 INSERT INTO ASSESSES VALUES (10, 13);

EDIT	APPL_ID	ESSN
	1	12
	2	13
	3	14
	4	12
	5	13
	6	14
	7	12
	8	13
	9	14
	10	13
row(s) 1 - 10 of 10		

▪ Table PUBLISHES:











INSERT INTO PUBLISHES VALUES (12, 1);  
 INSERT INTO PUBLISHES VALUES (12, 4);  
 INSERT INTO PUBLISHES VALUES (12, 7);  
 INSERT INTO PUBLISHES VALUES (13, 2);  
 INSERT INTO PUBLISHES VALUES (13, 5);  
 INSERT INTO PUBLISHES VALUES (13, 8);

INSERT INTO PUBLISHES VALUES (13, 10);

INSERT INTO PUBLISHES VALUES (14, 3);

INSERT INTO PUBLISHES VALUES (14, 6);

INSERT INTO PUBLISHES VALUES (14, 9);

EDIT	ESSN	RECR_ID
	12	1
	12	4
	12	7
	13	2
	13	5
	13	8
	13	10
	14	3
	14	6
	14	9
row(s) 1 - 10 of 10		

▪ Table SKILLS:

INSERT INTO SKILLS VALUES ('Accounting', 6);

INSERT INTO SKILLS VALUES ('Coordination', 11);

INSERT INTO SKILLS VALUES ('Customer Service', 17);

INSERT INTO SKILLS VALUES ('Data Analysis', 10);

INSERT INTO SKILLS VALUES ('Employee Relations', 13);

INSERT INTO SKILLS VALUES ('Financial Analysis', 15);

INSERT INTO SKILLS VALUES ('Graphic Design', 8);

INSERT INTO SKILLS VALUES ('Human Resource Management', 14);

INSERT INTO SKILLS VALUES ('IT Security', 5);

INSERT INTO SKILLS VALUES ('Legal Research', 21);

INSERT INTO SKILLS VALUES ('Marketing Strategy', 18);

INSERT INTO SKILLS VALUES ('Mechanical Engineering', 7);

INSERT INTO SKILLS VALUES ('Negotiation', 20);

INSERT INTO SKILLS VALUES ('Network Administration', 3);

INSERT INTO SKILLS VALUES ('Recruitment', 12);

INSERT INTO SKILLS VALUES ('Sales Techniques', 19);

INSERT INTO SKILLS VALUES ('Salesmanship', 9);







INSERT INTO SKILLS VALUES ('Software Development', 4);

INSERT INTO SKILLS VALUES ('Statistical Analysis', 16);

INSERT INTO SKILLS VALUES ('Strategic Planning', 2);

INSERT INTO SKILLS VALUES ('Web Development', 1);

EDIT	SKILLS	ESSN
	Accounting	6
	Coordination	11
	Customer Service	17
	Data Analysis	10
	Employee Relations	13
	Financial Analysis	15
	Graphic Design	8
	Human Resource Management	14
	IT Security	5
	Legal Research	21
	Marketing Strategy	18
	Mechanical Engineering	7
	Negotiation	20
	Network Administration	3
	Recruitment	12
row(s) 1 - 15 of 21 		

EDIT	SKILLS	ESSN
	Sales Techniques	19
	Salesmanship	9
	Software Development	4
	Statistical Analysis	16
	Strategic Planning	2
	Web Development	1
row(s) 16 - 21 of 21		

▪ Table PROVIDER:

INSERT INTO PROVIDER VALUES ('company A', 1);

INSERT INTO PROVIDER VALUES ('company A', 2);

INSERT INTO PROVIDER VALUES ('company A', 4);

INSERT INTO PROVIDER VALUES ('company A', 8);

INSERT INTO PROVIDER VALUES ('company A', 9);

INSERT INTO PROVIDER VALUES ('company A', 10);

INSERT INTO PROVIDER VALUES ('company A', 11);

INSERT INTO PROVIDER VALUES ('company A', 12);

INSERT INTO PROVIDER VALUES ('company A', 13);

INSERT INTO PROVIDER VALUES ('company A', 14);

INSERT INTO PROVIDER VALUES ('company A', 15);

INSERT INTO PROVIDER VALUES ('company A', 16);

INSERT INTO PROVIDER VALUES ('company A', 18);

INSERT INTO PROVIDER VALUES ('company A', 19);

INSERT INTO PROVIDER VALUES ('company A', 20);

INSERT INTO PROVIDER VALUES ('company A', 21);

INSERT INTO PROVIDER VALUES ('company A', 22);

INSERT INTO PROVIDER VALUES ('company A', 23);

INSERT INTO PROVIDER VALUES ('company A', 24);

INSERT INTO PROVIDER VALUES ('company A', 25);

INSERT INTO PROVIDER VALUES ('company A', 26);

INSERT INTO PROVIDER VALUES ('company A', 27);

INSERT INTO PROVIDER VALUES ('company B', 3);

INSERT INTO PROVIDER VALUES ('company C', 7);

INSERT INTO PROVIDER VALUES ('company D', 5);

INSERT INTO PROVIDER VALUES ('company D', 6);

EDIT	PROVIDER	BENF_ID
	company A	1
	company A	2
	company A	4
	company A	8
	company A	9
	company A	10
	company A	11
	company A	12
	company A	13
	company A	14
	company A	15
	company A	16
	company A	18
	company A	19
	company A	20
row(s) 1 - 15 of 26 		

EDIT	PROVIDER	BENF_ID
	company A	21
	company A	22
	company A	23
	company A	24
	company A	25
	company A	26
	company A	27
	company B	3
	company C	7
	company D	5
	company D	6
 row(s) 16 - 26 of 26		

## X. SQL QUERIES:

This part includes retrieving the data already inserted in 15 different and challenging complex scenarios.

1. Select first\_name and last\_name from the employee table where the employee doesn't have a dependent and also serves as a manager in a department.

```
SELECT e.fname AS first_name, e.lname AS last_name
FROM employee e
WHERE NOT EXISTS (
  SELECT 1
  FROM dependent d
  WHERE e.ssn = d.essn
) AND EXISTS (
  SELECT 1
  FROM department d
  WHERE e.ssn = d.mngr_ssn
);
```

**Results** Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME
youssef	borji
aleen	jaber
Jad	Al-Awar
amine	Abou chkra
khaled	Ahmad
Nabil	Haddad
Rita	Nasser

7 rows returned in 0.00 seconds [Download](#)



2. Identify employees within the accounting department who do not possess accounting skills, which might prompt further training or hiring considerations.

```
SELECT e.fname as first_name, e.lname as last_name
FROM employee e, department d
where e.d_id = d.dept_id
      and d.name = 'accounting'
      and NOT EXISTS (
        SELECT 1
        FROM skills s
        WHERE e.ssn = s.ssn
              and s.skills = 'Accounting'
      );
```

**Results** Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME
Khaled	Matar

1 rows returned in 0.00 seconds [Download](#)

3. Identify employees whose salaries are higher than the average salary within their respective departments.

```
SELECT distinct e.fname first_name, e.d_id as department_id, p.salary
FROM employee e, payroll p
WHERE p.salary > (
  SELECT AVG(p2.salary)
  FROM payroll p2
  WHERE p.ssn = e.ssn
)order by d_id;
```

**Results** Explain Describe Saved SQL History

FIRST_NAME	DEPARTMENT_ID	SALARY
Jad	1	25000
George	2	9000
Iouva	3	10500
Khaled	4	11000
Nabil	4	9200
youssef	5	10000
Sara	6	10000
Rita	7	9900
khaled	8	10000
aleen	9	8700
amine	10	10660

11 rows returned in 0.00 seconds [Download](#)

4. List all employees who have dependents and the total number of dependents each employee has.

```
SELECT e.ssn as id, e.fname as first_name, e.lname as last_name,  
       (SELECT COUNT(*)  
        FROM dependent  
        WHERE essn = e.ssn) as number_of_dependents  
FROM employee e  
WHERE EXISTS  
      (SELECT 1  
       FROM dependent  
       WHERE essn = e.ssn);
```

Results Explain Describe Saved SQL History

ID	FIRST_NAME	LAST_NAME	NUMBER_OF_DEPENDENTS
1	john	doe	2
3	mostapha	zein-eddine	1
4	George	Sabbagh	3
5	Sara	Dagher	6
14	louva	doe	3

5 rows returned in 0.01 seconds

[Download](#)

5. Select all applicants of applications studied by the manager of the human resources department.

```
SELECT a.name as applicant_name, a.app_id as applicant_id  
FROM applicant a, assesses ass  
WHERE a.app_id = ass.APPL_ID  
and ass.essn = (select e.ssn  
                from employee e, department d  
                where e.ssn = d.mngr_ssn  
                and d.name = 'Human Resources'  
);
```

Results Explain Describe Saved SQL History

APPLICANT_NAME	APPLICANT_ID
Karim Nassar	3
Rana Abi-Nader	6
Rami Harb	9

3 rows returned in 0.00 seconds

[Download](#)

6. List all employees whose family name starts with an 'A' and benefit from life insurance.

```
select distinct e.fname as first_name, e.lname as last_name
from employee e, benefits b
where e.lname like 'A%'
      and e.ssn = b.ssn
      and b.coverage_type = 'life insurance'
```

**Results** Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME
christine	Ahmad

1 rows returned in 0.01 seconds [Download](#)

7. List all managers who are late for work. (Shifts start at 9 and 5 minutes late are not considered late comings)

```
select e.fname, e.lname, a.time_in
from employee e, attendance a, department d
where a.time_in between '02-APR-24 09.05.00.000000' and '02-APR-24 09.30.00.000000'
      and e.ssn = a.ssn
      and d.mngr_ssn = e.ssn
```

**Results** Explain Describe Saved SQL History

FNAME	LNAME	TIME_IN
Sara	Dagher	02-APR-24 09.05.00.000000 AM

1 rows returned in 0.01 seconds [Download](#)

8. For each employee, determine the reward amount based on the following criteria:
  1. If the employee has 1 dependent, the reward amount is \$100.
  2. If the employee has 2 dependents, the reward amount is \$200.
  3. If the employee has 3 dependents, the reward amount is \$300.
  4. If the employee has 4 dependents, the reward amount is \$400.
  5. If the employee has more than 5 dependents, the reward amount is \$500.
  6. Else, the reward amount is \$0.

```
SELECT distinct e.fname as first_name, e.lname as last_name,
CASE
  WHEN (SELECT COUNT(dep.name) AS dc FROM dependent dep WHERE dep.eSSN = e.SSN) = 1 THEN '100$'
  WHEN (SELECT COUNT(dep.name) AS dc FROM dependent dep WHERE dep.eSSN = e.SSN) = 2 THEN '200$'
  WHEN (SELECT COUNT(dep.name) AS dc FROM dependent dep WHERE dep.eSSN = e.SSN) = 3 THEN '300$'
  WHEN (SELECT COUNT(dep.name) AS dc FROM dependent dep WHERE dep.eSSN = e.SSN) = 4 THEN '400$'
  WHEN (SELECT COUNT(dep.name) AS dc FROM dependent dep WHERE dep.eSSN = e.SSN) > 5 THEN '500$'
  else '0$'
END AS reward
FROM employee e;
```

FIRST_NAME	LAST_NAME	REWARD
aleen	jaber	0\$
Jad	Al-Awar	0\$
Nabil	Haddad	0\$
Khaled	Matar	0\$
Elie	Asmar	0\$
George	Sabbagh	300\$
Rita	Nasser	0\$
lina	el-halabi	0\$
john	doe	200\$
youssef	borji	0\$
louva	doe	300\$
Sara	Dagher	500\$
Lara	Hajj	0\$
Nour	Hajj	0\$
sara	el-halabi	0\$
khaled	Ahmad	0\$
christine	Ahmad	0\$
amine	Abou chkra	0\$
Tony	Antoun	0\$
Maya	Khair	0\$
mostapha	zein-eddine	100\$

21 rows returned in 0.01 seconds

[Download](#)

9. List all employees who have not taken any vacation.

```
SELECT e.FNAME as first_name, e.LNAME as last_name
FROM employee e
WHERE NOT EXISTS (
    SELECT 1
    FROM vacation v
    WHERE v.ESSN = e.SSN
);
```

**Results** Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME
mostapha	zein-eddine
youssef	borji
amine	Abou chkra
christine	Ahmad
Lara	Hajj
George	Sabbagh
louva	doe
lina	el-halabi
Maya	Khair
aleen	jaber
Jad	Al-Awar
Nour	Hajj
Nabil	Haddad

13 rows returned in 0.01 seconds

[Download](#)

10. Retrieve the number of employees on sick leave in each department, if any.

```
select d.name as department_name, count(*) as sick_leave
from employee e, department d, vacation v
where e.d_id = d.dept_id
      and e.ssn = v.essn
      and v.type = 'Sick Leave'
group by d.name;
```

**Results** Explain Describe Saved SQL History

DEPARTMENT_NAME	SICK_LEAVE
customer services	1
marketing	1

2 rows returned in 0.02 seconds [Download](#)

11. Identify all employees who have never been evaluated by anyone.

```
SELECT e.FNAME as first_name, e.LNAME as last_name
FROM employee e
WHERE NOT EXISTS (
  SELECT 1
  FROM evaluation ev
  WHERE ev.ESSN = e.SSN
);
```

**Results** Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME
mostapha	zein-eddine
Tony	Antoun
youssef	borji
sara	el-halabi
Lara	Hajj
George	Sabbagh
Iouva	doe
Rita	Nasser
Jad	Al-Awar
Nour	Hajj
Nabil	Haddad

11 rows returned in 0.00 seconds [Download](#)

12. Find employees who have dependents with a PHD degree.

```
SELECT DISTINCT e.FNAME as first_name, e.LNAME as last_name, d.name as dependent_name, d.relationship as relation
FROM employee e, dependent d
where e.SSN = d.ESSN
      and d.EDUCATION_LEVEL LIKE '%PHD%';
```

Results Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME	DEPENDENT_NAME	RELATION
mostapha	zein-eddine	fatima	wife

1 rows returned in 0.00 seconds [Download](#)

13. Identify employees who have salesmanship skills and have taken a vacation in 2024.

```
SELECT DISTINCT e.FNAME, e.LNAME
FROM employee e, skills s, vacation v
where e.SSN = s.ESSN
      and e.SSN = v.ESSN
      and s.SKILLS = 'Salesmanship'
      and v.START_DATE >= TO_DATE('2024-01-01', 'YYYY-MM-DD')
      and v.END_DATE <= TO_DATE('2024-12-31', 'YYYY-MM-DD');
```

Results Explain Describe Saved SQL History

FNAME	LNAME
Rita	Nasser

1 rows returned in 0.00 seconds [Download](#)



14. What are the evaluation details and feedback, for evaluations conducted by evaluator 15 and 18?

```
SELECT EVALUATOR_ID, EVAL_DATE, FEEDBACK
FROM EVALUATION
WHERE EVALUATOR_ID = 15
UNION
SELECT EVALUATOR_ID, EVAL_DATE, FEEDBACK
FROM EVALUATION
WHERE EVALUATOR_ID = 18;
```

**Results** Explain Describe Saved SQL History

EVALUATOR_ID	EVAL_DATE	FEEDBACK
15	04/02/2024	Performance below expectations, improvement needed in multiple areas.
18	04/02/2024	Consistently outstanding, a model employee.

2 rows returned in 0.00 seconds [Download](#)

15. What is the net salary distribution across different departments?

```
SELECT E.D_ID, D.NAME AS DEPARTMENT_NAME,
       SUM(P.SALARY + P.BONUS - P.DEDUCTION) AS TOTAL_NET_SALARY
FROM EMPLOYEE E, payroll p, department d
where E.SSN = P.ESSN
       and E.D_ID = D.DEPT_ID
GROUP BY E.D_ID, D.NAME
order by d_id;
```

**Results** Explain Describe Saved SQL History

D_ID	DEPARTMENT_NAME	TOTAL_NET_SALARY
1	Engineering	36700
2	Development	12975
3	Human Resources	20770
4	accounting	21025
5	finance	15220
6	IT	19720
7	customer services	15090
8	marketing	15278
9	sales	8750
10	legal	10760

10 rows returned in 0.01 seconds [Download](#)



## **XI. Normalization:**

All relations must undergo the normalization process after being created to assure satisfactory standards. The normalization process ranges from the first normal form to Boyce-Codd Normal Form. If a relation does not pass any of these normal forms, it is normalized into several relations.

### **I. First Normal Form:**

The standard form disallows the inclusion of multivalued attributes, composite attributes, or nested relations. To satisfy the requirements of the first Normal Form, every attribute must consist of individual and singular values. It is worth emphasizing that, inherently, all relations adhere to the first Normal Form because the relational model's design inherently prevents the presence of multivalued attributes, composite attributes, and nested relations.

### **II. Second Normal Form:**

A relation achieves second Normal Form when it meets the criteria of the first normal form, and each non-prime attribute in that relation is entirely functionally dependent on the primary key.

### **III. Third Normal Form:**

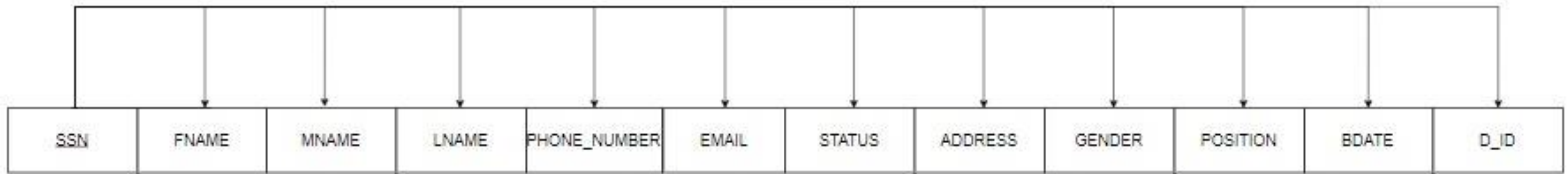
To meet the requirements of the third Normal Form, a relation must satisfy the conditions of the second normal form, ensuring that no non-prime attribute in the relation is transitively dependent on the primary key.

### **IV. Boyce-Codd Normal Form:**

The Boyce-Codd Normal Form, an extension of the third normal form, enhances the normalization process by ensuring that every determinant in all functional dependencies is exclusively a candidate key or a primary key.

## 1. EMPLOYEE:

FD1



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

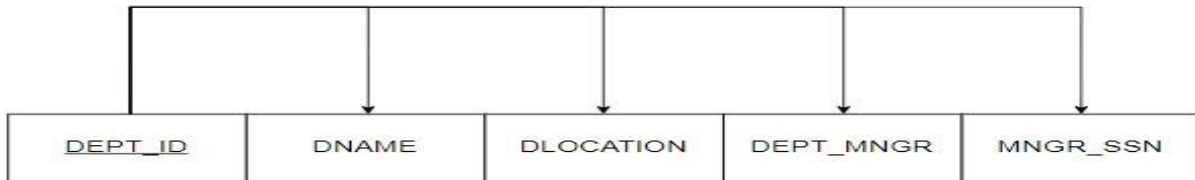
**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (SSN).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (SSN).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 2. DEPARTMENT

FD1



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

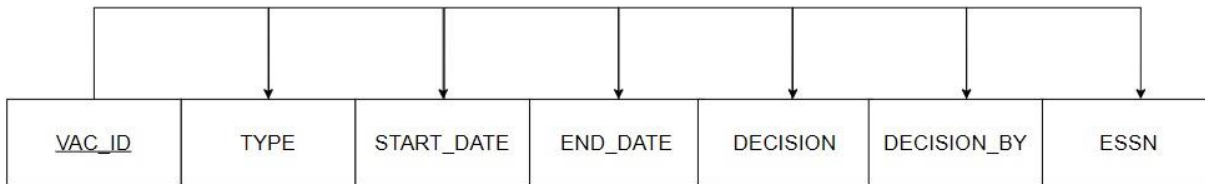
**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (DEPT\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (DEPT\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 3. VACATION

FD1



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

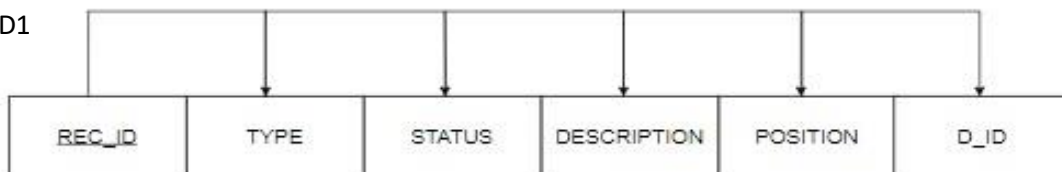
**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (VAC\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (VAC\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 4. RECRUTEMENT

FD1



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully

functionally dependent on the primary key (REC\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (REC\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 5. APPLICANT



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (APP\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (APP\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 6. EVALUATION



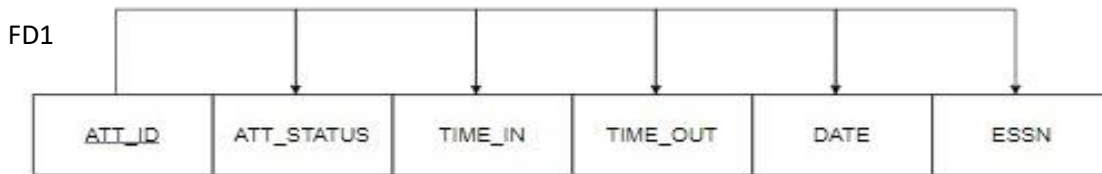
**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (EVAL\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EVAL\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 7. ATTENDANCE



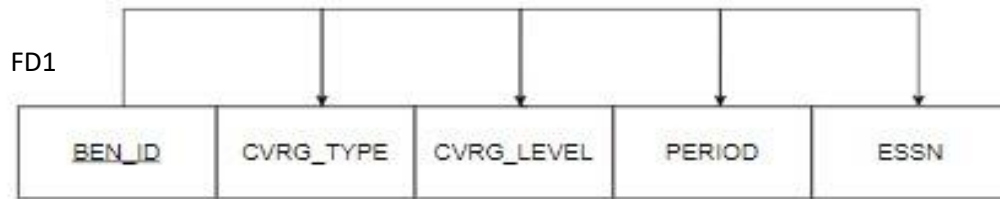
**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (ATT\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ATT\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 8. BENEFITS



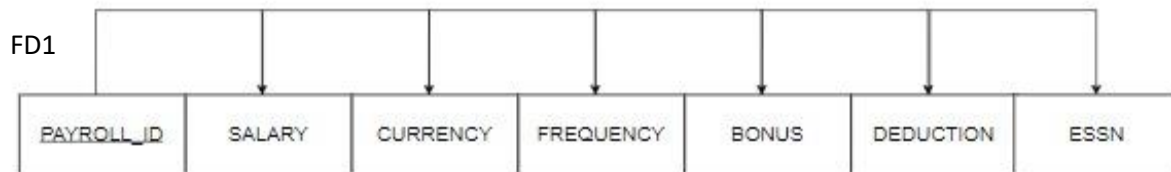
**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (BEN\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (BEN\_ID).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 9. PAYROLL



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

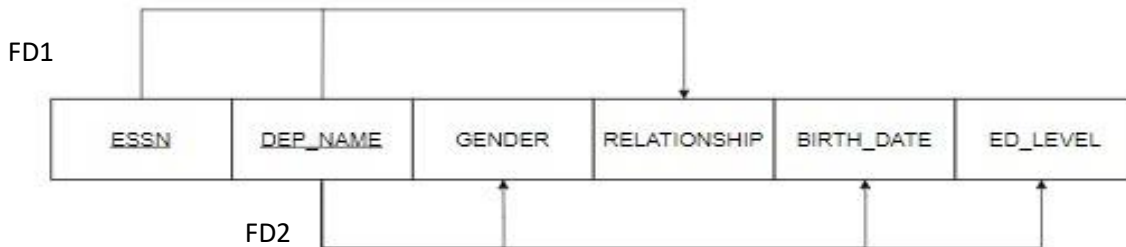
**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key (PAYROLL\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively

dependent on the primary key (PAYROLL\_ID).

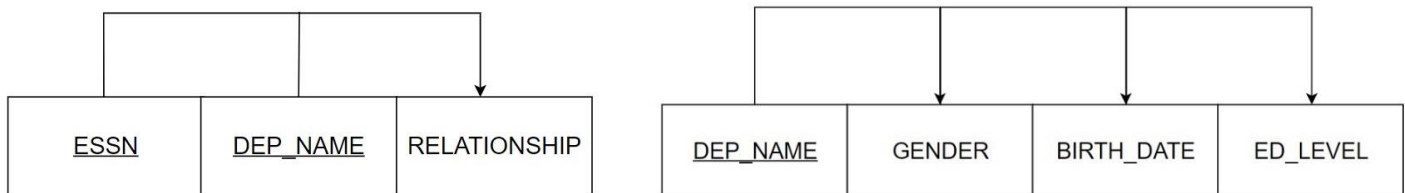
**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

#### 10. DEPENDENT



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation does not pass 2NF because gender, BIRTH\_DATE and ED\_LEVEL depend only on a part of the primary key (DEP\_NAME), so it must undergo 2NF normalization.



**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ESSN, DEP\_NAME).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 11. APPLIES



**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent on the primary key ({RECR\_ID, APPL\_ID}).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key ({RECR\_ID, APPL\_ID}).

**BCNF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 12. PUBLISHES



This relation passes all normal forms (**1NF, 2NF, 3NF, and BCNF**) because all of its attributes are composed of the primary key ({RECR\_ID, ESSN}).



### 13. ASSESSES

<u>APPL_ID</u>	<u>ESSN</u>
----------------	-------------

This relation passes all normal forms (**1NF, 2NF, 3NF, and BCNF**) because all of its attributes are composed of the primary key ({APPL\_ID, ESSN}).

### 14. SKILLS

<u>ESSN</u>	<u>SKILLS</u>
-------------	---------------

This relation passes all normal forms (**1NF, 2NF, 3NF, and BCNF**) because all of its attributes are composed of the primary key ({ESSN, SKILLS}).

### 15. PROVIDER

<u>BENF_ID</u>	<u>PROVIDER</u>
----------------	-----------------

This relation passes all normal forms (**1NF, 2NF, 3NF, and BCNF**) because all of its attributes are composed of the primary key ({BENF\_ID, PROVIDER}).

## **XII. Conclusion**

In the HRMS project meticulously crafted by Data Crafters, every aspect of data management and employee engagement was intricately woven together. From the meticulous collection of employee information to the skillful crafting of intuitive interfaces, Data Crafters left no stone unturned in ensuring the system's efficiency and effectiveness. With a keen eye for detail, they carefully tailored the database architecture to accommodate the diverse needs of HR operations. Through their expert craftsmanship, Data Crafters seamlessly integrated features for recruitment, payroll management, and performance evaluation, providing a comprehensive solution for human resource management. The HRMS project stands as a testament to Data Crafters' dedication to excellence in data-driven solutions, empowering organizations to navigate the complexities of workforce management with ease. Through their skillful craftsmanship, Data Crafters have woven together a robust HRMS that not only streamlines processes but also fosters employee satisfaction and organizational success.

### **XIII. Instructor's Feedback**

This page should be used by our instructor Dr. Abbas Bassam to write any comments, feedback, and suggestions for our database system. This page will be used to improve our upcoming report.