**Homework Assignment: Practicing Relationships between Tables**

**Task 1: Understanding Table Relationships**

**1. \*\*Defining Primary and Foreign Keys:\*\***

- Describe what a primary key is and provide an example:-

A primary key is simply a unique identifier for a row. And it is used as a to quickly parse data within the table. A table cannot have more than one primary key. A primary key’s main features are: It must contain a unique value for each row of data, it cannot contain null values and every row must have a primary key value.

Example:- Consider a table called “Students” that stores information about students in a school. Let’s assume we have the following attributes such as Student ID, Name, Age, and Grade. In this example, we will designate the “Student ID” as the primary key. The “Student ID” column serves as the primary key, uniquely identifying each student in the table.

**-** Describe what a foreign key is and provide an example:-

A foreign key is a set of attributes in a table that refers to the primary key of another table, linking these two tables. Foreign keys link together two or more tables in a relational database. These link two tables together. It is possible to have more than one foreign key in a table, and they can accept a null value. Foreign key values do not need to be unique; duplicate values can be stored in foreign key columns.

Example:- In a retail database system, there are two tables: Customers and Orders. The Customers table includes columns such as Customer ID, FirstName, LastName, Email, and Phone Number, with Customer ID serving as the primary key. The Orders table includes columns like Order ID, Order Date, Total Amount, and Customer ID, where Order ID is the primary key and Customer ID is a foreign key. The Customer ID in the Orders table establishes a link to the Customer ID in the Customers table, ensuring that each order is associated with a valid customer. This relationship means that for every order recorded in the Orders table, there must be a corresponding entry in the Customers table with the same Customer ID. This enforces referential integrity by ensuring that orders cannot exist without a corresponding customer, thus maintaining consistent and accurate data within the database. For example, if customer John Doe has a Customer ID of 101, any order he places will have 101 as the Customer ID in the Orders table, ensuring a clear and enforceable relationship between the two tables.

**2. \*\*Creating Related Tables:\*\***

Designing a database for a company: Departments and Employees

DEPARTMENTS TABLE:-

|  |  |  |  |
| --- | --- | --- | --- |
| DepartmentID | DepartmentName | ManagerID | Location |
|  |  |  |  |
|  |  |  |  |

In the below table “Departments” table DepartmentID is the Primary Key.

EMPLOYEES TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EmpolyeeID | Name | Email | HireDate | DepartmentID |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Foreign key**

In the above Employees table DepartmentID is foreign key matches theprimary key column in the `Departments` table.

**3. \*\*Populating Tables:\*\***

Three example records for the Department table:-

|  |  |  |  |
| --- | --- | --- | --- |
| **DepartmentID** | **DepartmentName** | **ManagerID** | **Location** |
| 100 | Human Resources | 1001 | Block A |
| 101 | Information Technology | 1002 | Block B |
| 102 | Marketing | 1003 | Block C |

**Primary Key**

In the above table DepartmentID is the Primary Key.

Five example records for the `Employees` table:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmpolyeeID** | **Name** | **Email** | **HireDate** | **DepartmentID** |
| 1001 | Ramya | abc@example.com | 2020-01-15 | 100 |
| 1002 | Vinay | vvcc@example.com | 2019-03-22 | 101 |
| 1003 | Navya | nuvyy@example.com | 2021-06-01 | 102 |
| 1004 | Sindhu | ssquare@example.com | 2018-11-05 | 103 |
| 1005 | Srilekha | sreeee@example.com | 2022-09-12 | 104 |

**Foreign key**

In the above table DepartmentID is the foreign key which matches the primary key column in the departments table.

**Task 2: Querying Related Tables**

**4. \*\*Exploring Table Relationships:\*\***

**-**How to retrieve a list of all employees along with their department names:-

To retrieve the list of all employees along with their department names we have to use SQL query that joins the Employees table with Departments table on the DepartmentID column.

**SELECT** Employees.EmployeeID, Employees.Name, Employees.Email, Employees.HireDate, Departments.DepartmentName

**FROM** Employees

**JOIN** Departments ON Employees.DepartmentID = Departments.DepartmentID;

- Finding all employees who have a matching department:-

Imagine departments as categories or groups within a company. Employees belong to specific departments based on their job function or area of expertise. Each employee has a "DepartmentID" attached to their information. To find employees with matching departments, we are required looking for employees who share the same label. Organize the employee table data by the DepartmentID. This creates groups of employees within the same department, making it easy to identify matches.

**5. \*\*Short Answer:\*\***

-Importance of establishing relationships between tables in a relational database:-

There are number of benefits by establishing relationships between tables,

1. It establishes a connection between a pair of tables that are logically related to each other.

2.It helps to minimize redundant data.

3.It enables you to fetch data from multiple tables simultaneously.

4.Ensures that there are no duplicates of data in the database.

-A situation where not having a primary or foreign key could cause issues in a database:-

Not having primary or foreign keys can severely compromise the integrity, consistency, and reliability of a database. On the other hand, when trying to insert an order with a non-existing Customer ID into an Orders table without a foreign key constraint, the insert will succeed, allowing the insertion of invalid data into the table. This can lead to data inconsistency and make it difficult to update or delete date.

**6. \*\*True or False:\*\***

- A foreign key can reference a primary key in the same table:- TRUE

- A primary key can contain duplicate values:- FALSE

**7. \*\*Scenario-Based Question:\*\***

If you want to add a new column `Manager\_ID` in the `Employees` table which references the `Employee\_ID` of another employee in the same table, describe how this new column can help in managing employee relationships within the organization. Provide an example:-

Adding a Manager\_ID column referencing existing Employees\_ID in the employees table creates a powerful tool for managing employee relationships. It establishes a clear reporting structure , simplifies task delegation and allows for generating organizational charts. By querying this data, we can gain insights into the overall company structure.

Example:-

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_ID | Name | Department | Manager\_ID |
| 101 | Ramya | Sales | NULL |
| 102 | Vinay | Marketing | 101 |
| 103 | Navya | Engineering | 101 |
| 104 | Sindhu | Sales | 102 |
| 105 | Sai | Engineering | 103 |

In the above example, Ramya has a Manager\_ID of NULL as she has no direct manager. Vinay and Navya report directly to Ramya as their Manager\_ID is 101. Sindhu reports to Vinay as her Manager\_ID is 102.Whereas, Sai reports to Navya as his Manager\_ID is 103 .

By querying the Employees table based on the Manager\_ID, we can retrieve a list of employees reporting to a specific manager (e.g., all employees who report to Vinay in Marketing). This data can be used for various purposes, such as sending targeted communication, assigning projects within teams, or analyzing team performance.