Case Study: Optimizing Non-Resident Payment Processing at the University of Kentucky

The University of Kentucky's Non-Resident Individual (NRI) Payment System is a specialized financial model designed to navigate the complexities of international tax regulations and university payment policies. By identifying administrative bottlenecks and leveraging structured data modeling, the university has created a solution that ensures legal compliance while streamlining disbursements to international students, employees, and guests.

Problem 1: Navigating Complex Tax Classifications

Real-World Challenge: Determining the correct taxpayer status for non-resident individuals is a significant challenge due to varying visa types, international tax treaties, and federal vs. state requirements. Without a centralized system, inconsistent status determination can lead to legal penalties or incorrect tax withholding.

The Solution: The system introduces a structured **Department Administrator** entity responsible for determining the taxpayer status of each NRI and assisting with mandatory form completion. By establishing a **1:N relationship** between administrators and NRIs, the university ensures that every individual has a dedicated official overseeing their tax profile. This specialized oversight bridges the gap between complex government regulations and individual payment requests.

Problem 2: Fragmentation of Payment Types

Real-World Challenge: Payments to non-residents are not uniform; they range from professional services and awards to travel reimbursements. Originally, managing these as separate, disconnected entities created data redundancy and made it difficult to generate a single Payment Request Document (PRD).

The Solution: The design utilizes an **Enhanced Entity-Relationship (EER)** approach by creating a **PRD superclass** with a **disjoint constraint**. This allows the system to categorize payments into four mutually exclusive subclasses: **Service, Non-Service, Awards, and Guest Expenses**. This architecture ensures that while each payment type retains its unique requirements (like specific 8233 or BPM E-7-7 forms), they all feed into a unified document structure for final reporting.

Problem 3: Documentation and Audit Trails

Real-World Challenge: Attaching critical physical documentation, such as visa copies and I-94 forms, to financial records often leads to "information silos" where supporting evidence is separated from the transaction data.

The Solution: To mitigate complexity, the system integrates documentation directly into the **PRD entity**. By including attributes for visa copies, invoice reasoning, and taxpayer identification numbers directly within the request document, the system creates a self-contained

audit trail. This ensures that any processed **Payment** is uniquely identified by a transaction_id and backed by a complete, attached PRD record.

Top Features of the NRI Payment System:

- **Department Administrator Oversight**: Official assigned to uniquely identify NRIs by Emp_id, determine tax status, and route forms to the IRS.
- **NRI Profiles**: Comprehensive tracking of non-residents via Visa_no, type, I-94, and Mailing Address to ensure accurate identification.
- **Unified PRD System**: A centralized Payment Request Document that serves as the superclass for all specialized payment types, containing descriptions, invoices, and visa copies.
- **Automated Tax Handling**: Integrated tracking of federal and state taxes, as well as tax_exemption status directly within the Payment entity.
- **Disjoint Payment Subclasses**: Specific entities for Service, Non-Service, Awards, and Guest Expenses that enforce specific form requirements (e.g., 8233 or BPM E-7-7) depending on the nature of the payment.
- **APS Integration**: The Account Payable Services entity (identified by Branch_id) provides central oversight for the final distribution of payments.

Based on the structure of the provided case study format, the **Relational Model** (the diagram showing tables with specific attributes and linked arrows) should be included as a new section immediately following the **Schema Description** and just before the **ER Diagram** section.

This placement allows the reader to first understand the textual definitions of the entities, then see how they are structured as tables (Relational Model), and finally view the high-level conceptual relationships (ER/EER Diagram).

Here is the updated schema description incorporating the specific details from your attached relational model and introduction.

Schema Description: University of Kentucky Non-Resident Payment System
The schema for the University of Kentucky's Non-Resident Individual (NRI) Payment System involves multiple entities designed to track payments while ensuring policy compliance. These entities include administrators, individuals, and various specialized payment request documents (PRD).

Department Administrator Entity

Administrators are responsible for managing NRIs within their departments.

- **Emp_id (Primary Key):** A unique identifier for the department administrator.

- **FName:** The first name of the administrator.
- **LName:** The last name of the administrator.

Employed Non-Resident Individual (NRI) Entity

NRIs are the core recipients of the payments managed by the system.

- **Visa_No (Primary Key):** A unique identifier based on the individual's visa documentation.
- **TIN:** The Taxpayer Identification Number for the individual.
- **Emp_id (Foreign Key):** Links the NRI to a specific Department Administrator for identification.
- **FName / LName:** The full name of the non-resident individual.
- **Visa_Type:** The specific visa classification held by the individual.
- **Address / Country:** The mailing address and country of origin for the individual.

Payment Entity

This entity tracks the specific financial disbursements made to individuals.

- **Transaction_id (Primary Key):** A unique identifier for each financial transaction.
- **Visa_No (Foreign Key):** Connects the payment to a specific NRI.
- **Amount:** The total monetary value of the disbursement.
- **Federal_Tax / State_Tax:** The amount of tax withheld for federal and state requirements.
- **Tax_Exemption:** Information regarding any applied tax exemptions.

PRD (Payment Request Document) Entity

The PRD serves as a superclass for various types of payment requests.

- **PRD_ID / Transaction_id (Primary Key):** A unique identifier linked to the specific payment transaction.
- **Invoice_No:** The unique invoice number associated with the request.
- **Visa_No (Foreign Key):** Identifies which NRI the request is for.
- **Description:** A summary of the payment's purpose.
- **FName / LName / Address / Visa_Type:** Captured documentation details used for verification.

PRD Subclasses (Specialization)

Under a **disjoint constraint**, a PRD can belong to exactly one of the following specialized subclasses:

- **Service:** Requests for labor-based payments, including an **8233_Form**.
- **Non_Service:** General payments requiring a **Vendor_App**, **TIN**, and **8233_Form**.
- **Awards:** Accolades or prize payments categorized by a specific **Category**.

- **Guest_Expenses:** Reimbursement for visitors, specifically requiring **BPME-7-7** documentation.

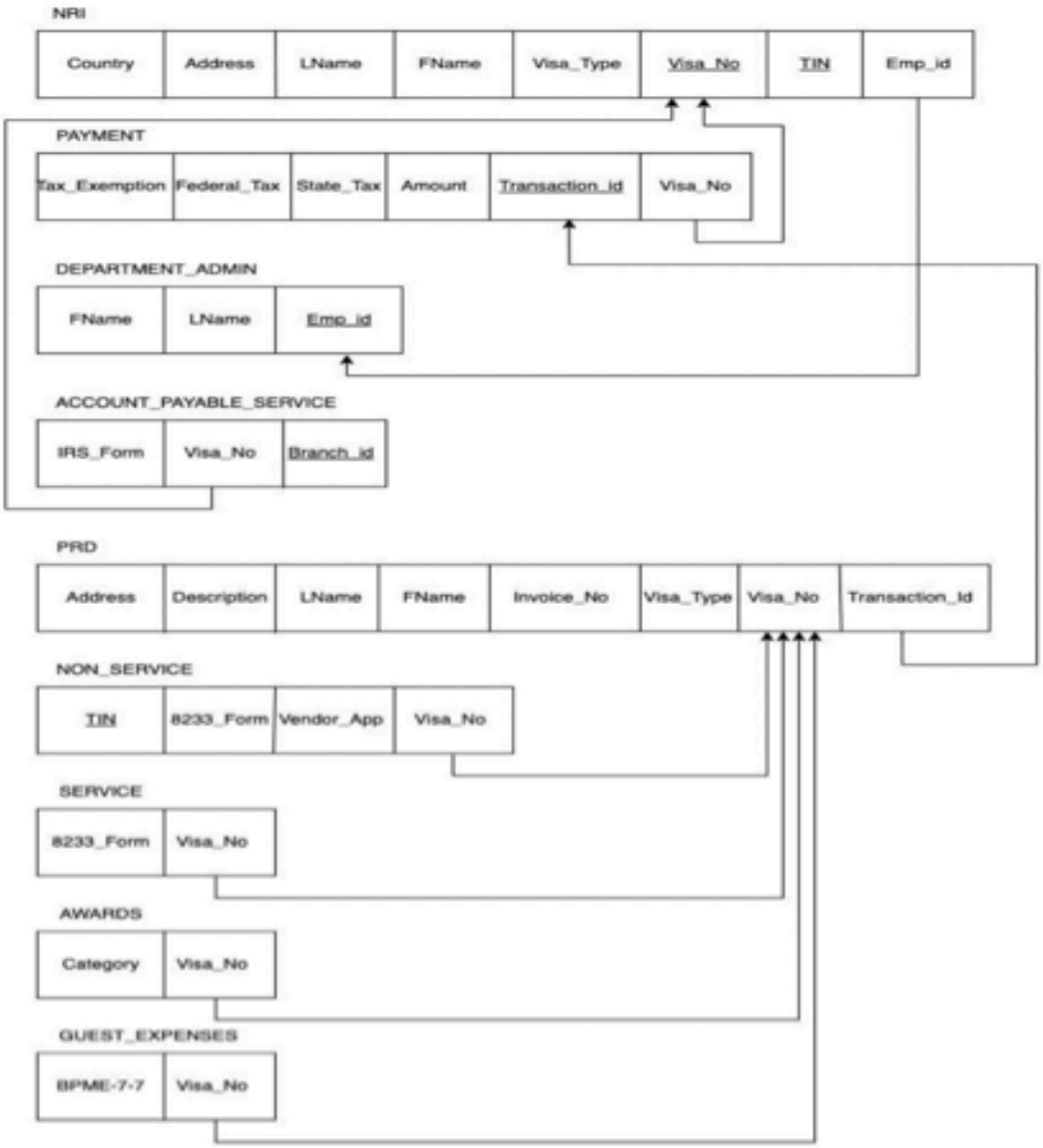
Relationships

- **Admin manages NRI:** One administrator can determine the tax status for multiple NRIs (1:N).
- **NRI receives Payment:** A single NRI can receive multiple payments over time (1:N).
- **Payment contains PRD:** Each payment transaction is associated with one specific PRD (1:1).
- **PRD Specialization:** The PRD is specialized into four disjoint subclasses: Service, Non-Service, Award, or Guest Expenses.

Relational Model:
The relational model represents the database schema as a series of interrelated tables. It defines the specific data types, primary keys (used to uniquely identify records), and foreign keys (used to link tables together). In this model, the University of Kentucky payment system is normalized to ensure data integrity and to support the disjoint constraint on payment types.

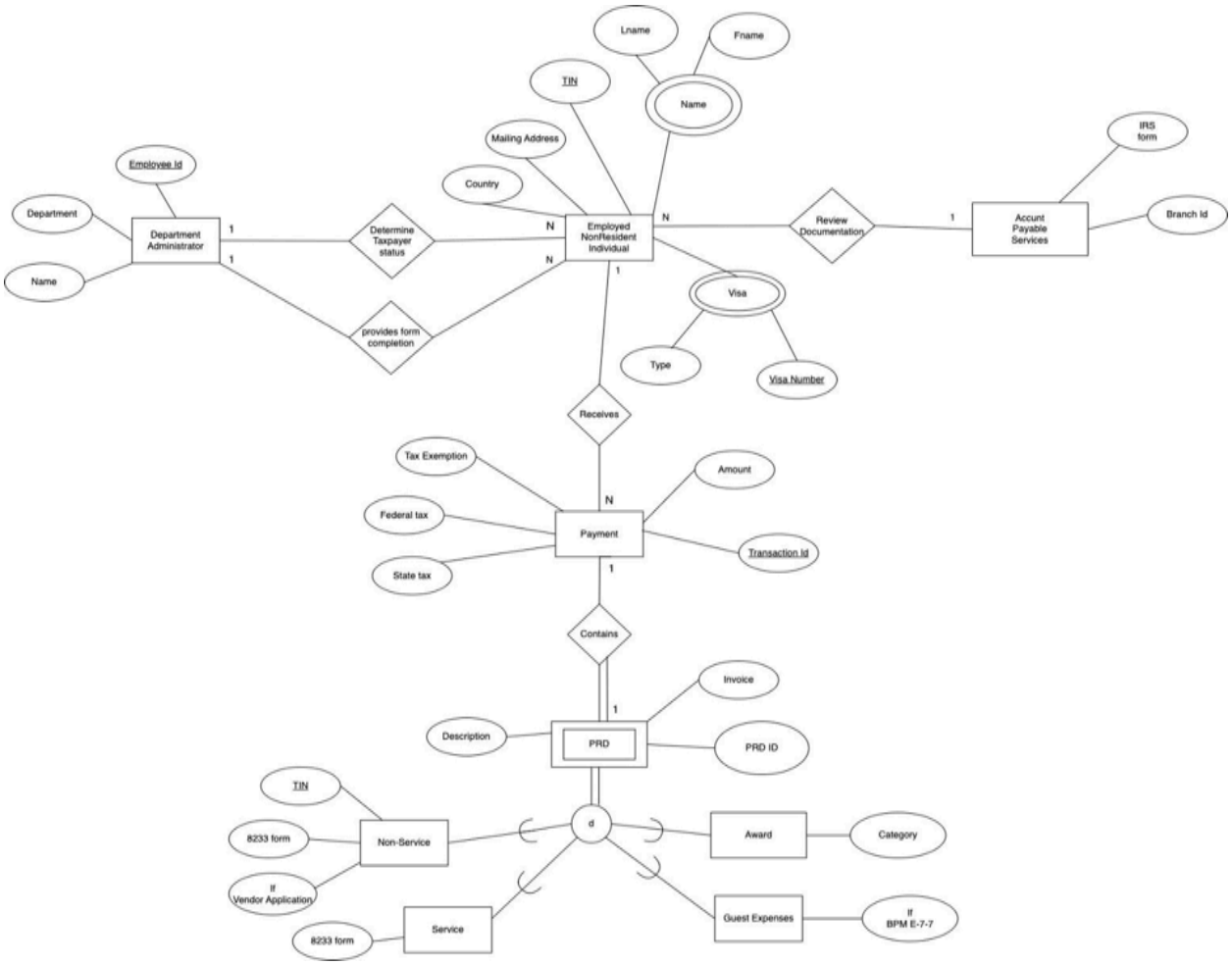
Table Structures:

- **NRI Table**: Contains the core attributes of the non-resident individual, including `Visa_No` (Primary Key), `TIN`, `FName`, `LName`, `Visa_Type`, `Address`, `Country`, and `Emp_id` (Foreign Key).
- **Payment Table**: Records disbursement details with `Transaction_id` (Primary Key) and `Visa_No` (Foreign Key).
- **Department_Admin Table**: Stores administrative data with `Emp_id` (Primary Key), `FName`, and `LName`.
- **Account_Payable_Service Table**: Manages document review with `Branch_id` (Primary Key), `IRS_Form`, and `Visa_No` (Foreign Key).
- **PRD Table**: The master table for payment requests, containing `Transaction_id` (Primary Key/Foreign Key), `Invoice_No`, `Description`, and individual contact details.
- **Specialized Tables (Subclasses)**: The tables **Service**, **Non_Service**, **Awards**, and **Guest_Expenses** all use `Visa_No` as a linking attribute to the PRD and NRI entities to maintain the disjoint relationship.



EER Diagram:

The provided EER diagram visually depicts the structure of this model, highlighting the specialization of the PRD entity into its four distinct subclasses through a disjoint constraint. This visual representation ensures a clear understanding of the mandatory attributes for different payment types while maintaining the integrity of the core relationship between the individual, the department, and the central accounting services.



Conclusion:

The University of Kentucky's database model transforms a high-risk administrative hurdle into a structured, transparent process. By fostering clear accountability through Department Administrators, utilizing EER specialization for payment types, and ensuring robust data-to-document mapping, the system effectively addresses the challenges of international financial compliance. This user-centric approach ensures that the university can fulfill its financial obligations to non-residents accurately and efficiently.

SQL Data Definition Language (DDL) and data query sql scripts to actually create these tables in a database

```
CREATE TABLE Account_Payable_Service(

IRS_form nvarchar(50) NULL,

Fk_Visa_No int,

PK_Branch_Id int PRIMARY KEY,
```

```
FOREIGN KEY(Fk_Visa_No) REFERENCES NRI(Visa_No)
);

CREATE TABLE Awards(
Category nvarchar(50) NULL,
Fk_PRD_Id int NULL,
FOREIGN KEY(Fk_PRD_Id) REFERENCES PRD(Pk_prd_Id)
);

CREATE TABLE Department_Admin(
Fname nvarchar(50) NULL,
Lname nvarchar(50) NULL,
PK_Emp_Id int PRIMARY KEY
);

CREATE TABLE Guest_Expenses(
BPME_77 nvarchar(50) NULL,
Fk_PRD_Id INT,
Foreign KEY(Fk_PRD_Id)
REFERENCES PRD(Pk_PRD_Id)
);

CREATE TABLE Non_Service(
TIN int NULL,
Form_8233 nvarchar(50) NULL,
Vendor_Application nvarchar(50) NULL,
Fk_PRD_Id INT,
FOREIGN KEY(Fk_PRD_Id) REFERENCES PRD(Pk_PRD_Id)
);
```

```
CREATE TABLE NRI(
Fname nvarchar(50) NULL,
Lname nvarchar(50) NULL,
Country nvarchar(50) NULL,
Address nvarchar(50) NULL,
Visa_Type nvarchar(50) NULL,
Pk_Visa_No int PRIMARY KEY,
TIN int NULL,
Fk_Emp_Id int NULL,
FOREIGN KEY(Fk_Emp_Id) REFERENCES Department_Admin(pk_Emp_Id)
);

CREATE TABLE Payment(
Tax_Exemption nvarchar(50) NULL,
Federal_Tax nvarchar(50) NULL,
State_Tax nvarchar(50) NULL,
Amount int NULL,
Pk_Transaction_Id int PRIMARY KEY,
Fk_Visa_No int NULL,
FOREIGN KEY(Fk_Visa_No) REFERENCES NRI(Pk_Visa_No)
);


CREATE TABLE PRD(
Description varchar(50) NULL,
Invoice_No int NULL,
Pk_PRD_Id int PRIMARY KEY,
Fk_Transaction_Id int NULL,
```

```
FOREIGN KEY(Fk_Transaction_Id) REFERENCES
Payment(Pk_Transaction_Id));
```

```
CREATE TABLE Service(
Form_8233 nvarchar(50) NULL,
PRD_Id int NULL,
FOREIGN KEY(PRD_Id) REFERENCES PRD(Pk_PRD_Id)
);
```

INSERT INTO "NRI" VALUES (Fname, Lname, CountrY, Address,Visa_Type, Visa_No(P), TIN, Emp_Id(f-DEP ADM);

INSERT INTO "NRI" VALUES (‘James’,’Hardy’, 'Germany’, '1009 South First Street','F1’, ‘Z1278’, 12345,'1’);

INSERT INTO "NRI" VALUES (‘Tom’,’Cruise’, 'Spain’, ‘508 East Daniel Street','F1’, ‘K2349’, 12345,'1’);

INSERT INTO "NRI" VALUES (‘Sri’,’GovindaRaju’, 'India’, ‘519 East Daniel Street','F1’, ‘W1299’, 12345,'2’);

INSERT INTO " Department_Admin " VALUES (Fname, Lname, Emp_Id(p));

INSERT INTO " Department_Admin " VALUES (‘Bob’,’Danielson’,'1’);

INSERT INTO " Department_Admin " VALUES (‘Ricky’,’Martin’,'1’);

INSERT INTO " Account_Payable_Service " VALUES (IRS_form

,Visa_No(F-NRI), PK_Branch_Id);

INSERT INTO " Payment " VALUES (Tax_Exemption, Federal_Tax, State_Tax, Amount, Pk_Transaction_Id(P), Fk_Visa_No(F-NRI));

INSERT INTO " PRD " VALUES (Description, Invoice_No,Pk_PRD_Id(P)

, Fk_Transaction_Id (f-payment));

INSERT INTO " Service" VALUES (Form_8233,PRD_Id(F-PRD));

INSERT INTO " Non_Service " VALUES (TIN,Form_8233, Vendor_Application, Pk_PRD_Id((F-PRD)))

INSERT INTO " Guest_Expenses " VALUES (Description,Invoice_No,Pk_PRD_Id(P))

INSERT INTO " Awards" VALUES (Category,PRD_Id(F))

INSERT INTO " Payroll_Service " VALUES (Pay_Id, Dept_No, Location, Fk_Branch_Id)(Account_Payable_Service))

Queries :

1) **Query the names of the Non Resident Individuals who is eligible for tax exemptions.Also attach their PRD descriptions and their 8233 Forms? --> D**

```
Select n.Fname + n.Lname as Fullname ,pd.Description from NRI n join
Payment p on n.Visa_No=p.Visa_No
join PRD pd on p.Transaction_Id=pd.Transaction_Id
join Service s on pd.PRD_Id=s.PRD_Id
where p.Tax_Exemption= 'Yes' and s.[8233_form]='yes';
```

2) Query the names of department administrators who assisted Indian NRI'S with the completion of the forms that are in compliance of compliance with Internal Revenue Service (IRS) regulations

```
select d.Fname + ' ' + d.Lname as Fullname from Department_Admin d
where d.Emp_Id IN
(Select Emp_Id from NRI n where n.Country = 'India')
order by Emp_Id;
```

3) Query the names of NRIs who is eligible for tax exemptions and received awards under rewards category ?

```
Select n.Fname + n.Lname as Fullname from NRI n join
Payment p on n.Visa_No=p.Visa_No
join PRD pd on p.Transaction_Id=pd.Transaction_Id
join Awards a on pd.PRD_Id=a.PRD_Id
where p.Tax_Exemption= 'Yes' and a.Category='Rewards';
```

4) Write a query for calculating the total number of NRIs with 'F1' visa status?

```
select COUNT(n.Visa_No) as Total_NRI from NRI n join payment p on
n.Visa_No = p.Visa_No
where n.Visa_Type = 'F1'
```

THE END