BearcatWealth
SMART FINANCE. STRONG GROWTH.

Employee Travel and Expense Database

Database Implementation and Application: Final Report

GROUP MEMBERS:

Olga Auza
olga.rodriguez2@baruchmail.cuny.edu

Geraldine Hirschhorn
geraldine.hirschhorn@baruchmail.cuny.edu

Shreya Karki
shreya.karki@baruchmail.cuny.edu

Wenli Li
wenli.li1@baruchmail.cuny.edu

Amy Sukhoo
amy.sukhoo@baruchmail.cuny.edu

David Zhai
david.zhai@baruchmail.cuny.edu

Course Number: CIS 9340
Section: UTA

BUSINESS DESCRIPTION

BearcatWealth is a local finance firm with a large network of clients in the US. The company prides itself on building and developing lasting relationships with its clients. Employees often travel to different regions in the US to meet with clients, a vital expense necessary for BearcatWealth's day-to-day operation, representing the company in key business transactions. In addition to travel, employees host client lunch meetings, attend conferences, and entertain clients in less formal settings to strengthen these relationships. However, managing these expenses has become a challenge, so simplifying the process has become necessary.

PROBLEM STATEMENT & PROPOSED SOLUTIONS

BearcatWealth currently lacks a centralized database system for managing travel and expense reporting, leading to operational inefficiencies, compliance risks, and limited financial oversight. The existing manual process has caused confusion between Departments, delayed reimbursement, and increased the chances of lost receipts. Additionally, without real-time budget tracking or a structured approach to vendor management, the company struggles to maintain transparency and enforce expense policies effectively.

To address these issues, BearcatWealth seeks to implement a comprehensive solution that simplifies travel requests, automates expense tracking, and ensures efficient reimbursements. The proposed system will enhance transparency across Departments, improve compliance with company policies, and provide valuable insights into company operations.

The key problems and corresponding solutions are outlined below:

Problems

- **Delayed reimbursements**: Manual expense submissions often result in late reporting, causing significant delays in reimbursement processing.
- **Lost or missing receipts**: Reliance on paper receipts increases the likelihood of misplaced documentation, complicating expense validation and audit processes.
- **Lack of policy enforcement**: Without automated checks, expenses that do not comply with company policies may be approved, leading to budget overruns and financial discrepancies.
- **Unclear travel budgets**: The absence of real-time budget tracking makes it difficult to monitor allocated versus spent travel budgets, potentially resulting in overspending.
- **Manual approval process**: Managers spend substantial time manually reviewing expenses, reducing overall productivity and slowing down the approval cycle.
- **Vendor and payment management issues**: Inefficient tracking of payments to travel agencies, airlines, and hotels creates reconciliation challenges and increases the risk of duplicate payments.

Solutions

- **Electronic expense submission**: Implementing an electronic submission system with automatic receipt uploads will streamline the process, eliminating delays and reducing the risk of missing receipts.
- **Automated policy compliance oversight**: The system will incorporate automated validation to ensure expenses comply with company policies, flagging non-compliant submissions before processing.
- **Real-Time budget monitor**: By tracking allocated versus spent travel budgets in real-time, the system will provide enhanced financial control and prevent overspending.
- **Automated approval workflows**: Automation of the approval process will reduce the administrative burden on managers, accelerating the review cycle and improving efficiency.
- **Integrated vendor management:** The solution will streamline vendor management by tracking all payments to travel agencies, airlines, and hotels, ensuring accurate reconciliation and eliminating duplicate payments.

INFORMATION NEEDS

To create a central system for managing travel and expense reporting across the Departments in BearcatWealth, a database is needed to track the current budget and ensure the travel and meal expenses are reviewed and approved promptly.

The system will store essential internal information such as the employee's and their manager's contact information, as well as the Department in which they work. An employee's travel request should include the EmployeeID, SupervisorID, Destination, Dates, Purpose, Estimated Cost, and Status and Submission Date. This information offers visibility into company-wide travel and provides adequate contact details for follow-ups on travel inquiries.

Depending on the Department's allocated funds, the employee's travel request will be reviewed and approved. The system will also match requested travel dates to travel vendors and their prices to utilize the budget effectively.

Additionally, there will be live tracking of the approval process for each employee organized by Department. Approval information includes ApprovalID, Approval Date, Rejection Date, Status, and Rejected Reason. Part of the approval process is ensuring the travel spending complies with company policy.

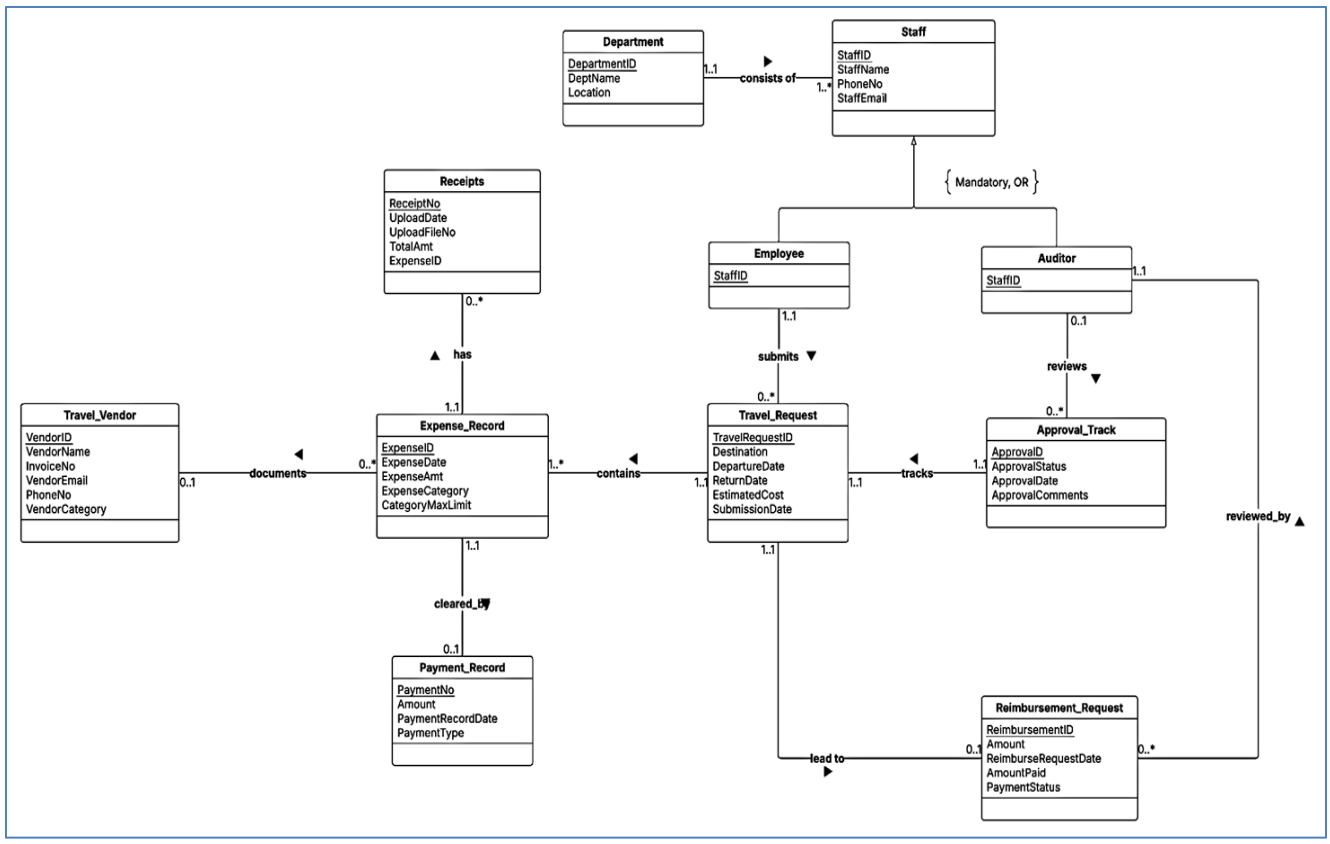
Once approved, expenses will be recorded based on their categories based on meals, airfare, hotel purchases, and purchase dates from the different vendors. All purchases will require receipts to be uploaded into the system with the Receipt Number and the ExpenseID recorded.

The system will also monitor corporate card usage, and tracking details such as CardID, EmployeeID, Expiry Date, and Credit Limit to maintain spending control before, during, and after the travel duration. All digital copies of receipts, account balances, and expense records will be available in the system at all times.

To streamline payments and vendor management, the system will track vendors, and their categories such as flight, hotel, transportation and contact details. Payments for reimbursements and company expenses will be recorded, to ensure accurate financial tracking. To enforce compliance with spending limits, the system will also integrate expense policy defining the maximum allowable amount for the different categories such as travel, meals, and transportation.

The system tracks receipts uploaded by employees, organized by file number. Employees must upload receipts for all expenses to verify and enforce limits. Receipts are classified into meals, transportation, or hotels. Receipts are automatically checked against policy limits to keep Departments within budget, helping prevent overspending, ensure compliance, and streamline approvals while maintaining accurate records.

ERD MODEL



RELATIONSHIP SENTENCES

Department → Staff
One Department consists of one or many Staff. 1..*
A Staff member can belong to one and only one Department. 1..1

Staff→ Employee
Staff → Auditor
Every Staff must be either an Employee or an Auditor, but not both

Employee → Travel_Request
One Employee may submit 0 to many Travel_Requests 0..*
One Travel_Request belongs to one Employee. 1..1

Auditor → Approval_Track
One Auditor may review zero or many Approval_Track. 0..*
One Approval_Track must be reviewed by one Auditor. 1..1

Auditor → Reimbursement_Request
One Auditor may review zero or many Reimbursement_Request. 0..*
One Reimbursement_Request must be reviewed by one Auditor. 1..1

Approval_Track → Travel Requests
Each Approval_Track must track one Travel_Request. 1..1
Each Travel_Request must be tracked by one Approval_Track. 1..1

Travel_Request → Expense_Record
One Travel_Request can contain one or more expense records. 1..*
One Expense_Record belongs to only one Travel_Request. 1..1

Travel_Request →Reimbursement_Request
One Travel_Request may lead to one Reimbursement_Request 0..1
One Reimbursement_Request connects with one Travel_Request. 1..1

Expense_Record → Receipt
One Expense_Record has zero or more receipts. 0..*
One Receipt belongs to exactly one Expense Record. 1..1

Expense_Record → Travel_Vendor
One Expense_Record documents zero or one Travel_Vendor. 0..1
One Travel_Vendor can be documented in zero or more Expense_Record. 0..*

Expense_Record→ Payment_Record
One payment_record clears only one expense record. 1..1
One expense record may be cleared by zero or one payment record. 0..1

RELATIONAL MODEL

- Travel_Vendor (VendorID, VendorName, InvoiceNo, VendorEmail, PhoneNo, VendorCategory)
- Expense_Record (ExpenseID, ExpenseDate, ExpenseAmt, ExpenseCategory, CategoryMaxLimit, VendorID(fk), TravelRequestID(fk))
- Receipts (ReceiptNo, UploadDate, UploadFileNo, TotalAmt, ExpenseID(fk))
- Payment_Record (PaymentNo, Amount, PaymentRecordDate, PaymentType,ExpenseID(fk))
- Department (DepartmentID, DeptName, Location)
- Staff (StaffID, StaffName, PhoneNo, StaffEmail, DepartmentID(fk))
- Auditor (StaffID, StaffName, PhoneNo, StaffEmail)
- Employee (StaffID, StaffName, PhoneNo, StaffEmail)
- Travel_Request(TravelRequestID, Destination, DepartureDate, ReturnDate, EstimatedCost, SubmissionDate, StaffID(fk))
- Reimbursement_Request (ReimbursementID, Amount, ReimbursementRequestDate, AmountPaid, PaymentStatus, TravelRequestID(fk), StaffID(fk))
- Approval_Track (ApprovalID, ApprovalStatus, ApprovalDate, ApprovalComments, StaffID(fk))

NORMALIZATION

Travel_Vendor Table:

FD1: VendorID → VendorName, InvoiceNo, VendorEmail, PhoneNo, VendorCategory

- **Step 1**: Primary key? Yes, VendorID
  - **1NF Complete**
- **Step 2:** 2NF test. Any partial dependency? No
  - **2NF Complete**
- **Step 3**: 3NF test. Any transitive dependency? No
  - **3NF Complete**
- **Step 4**: BCNF test. Are any determinants not candidate keys? No
- **Normalize to BCNF result:**
  - **R1:** (VendorID, VendorName, InvoiceNo, VendorEmail, PhoneNo, VendorCategory)

Expense_Record Table:

FD1: ExpenseID → ExpenseDate, ExpenseAmt, ExpenseCategory, CategoryMaxLimit, VendorID, TravelRequestID
FD2: ExpenseCategory → CategoryMaxLimit (Each ExpenseCategory is associated with a fixed CategoryMaxLimit

- **Step 1:** Primary key? Yes, ExpenseID
  - **1NF Complete**
- **Step 2**: 2NF test. Any partial dependency? No
  - **2NF Complete**
- **Step 3**: 3NF test. Any transitive dependency? Yes,

    - ExpenseCategory → CategoryMaxLimit
    - Splitting:
        - R1(ExpenseID, ExpenseDate, ExpenseAmt, ExpenseCategory, VendorID, TravelRequestID)
        - R2(ExpenseCategory, CategoryMaxLimit)
        - **3NF Complete**
- **Step 4:** BCNF test. Are any determinants not candidate keys? No
- **Normalize to BCNF result:**
    - **R1:** (ExpenseID, ExpenseDate, ExpenseAmt, ExpenseCategory, VendorID, TravelRequestID)
    - **R2**: (ExpenseCategory, CategoryMaxLimit)

Receipts Table:

FD1: ReceiptNo→ UploadDate, UploadFileNo, TotalAmt, ExpenseID

- **Step 1:** Primary key? Yes, ReceiptNo
    - **1NF Complete**
- **Step 2:** 2NF test. Any partial dependency? No
    - **2NF Complete**
- **Step 3:** 3NF test. Any transitive dependency? No
    - **3NF Complete**
- **Step 4:** BCNF test. Are any determinants not candidate keys? No
- **Normalize to BCNF result:**
    - **R1:** (ReceiptNo, UploadDate, UploadFileNo, TotalAmt, ExpenseID)

Payment_Record Table:

FD1: PaymentNo → Amount, PaymentRecordDate, PaymentType, ExpenseID

- **Step 1:** Primary key? Yes, PaymentNo
    - **1NF Complete**
- **Step 2**: 2NF test. Any partial dependency? No
    - **2NF Complete**
- **Step 3**: 3NF test. Any transitive dependency? No
    - **3NF Complete**
- **Step 4**: BCNF test. Are any determinants not candidate keys? No
- **Normalize to BCNF result:**
    - **R1**: (PaymentNo, Amount, PaymentRecordDate, PaymentType, ExpenseID)

Department Table:

FD1: DepartmentID → DeptName, Location

- **Step 1**: Primary Key? Yes, DepartmentID
    - **1NF Complete**
- **Step 2:** 2NF Partial-key dependencies? No

    - **2NF Complete**
- **Step 3:** 3NF Transitive dependencies? No
    - **3NF Complete**
- **Step 4**: BCNF Non-candidate Keys? No
    - **BCNF Complete**
- **Relations:**
    - **R1**: (DepartmentID, DeptName, Location)

Staff Table:

FD1: StaffID → StaffName, PhoneNo, StaffEmail, DepartmentID

- **Step 1:** Primary Key? Yes, StaffID
    - **1NF Complete**
- **Step 2:** 2NF Partial-key dependencies? No
    - **2NF Complete**
- **Step 3:** 3NF Transitive dependencies? No
    - **3NF Complete**
- **Step 4:** BCNF Non-candidate Keys? No
    - **BCNF Complete**
- **Relations:**
    - **R1**:(StaffID, StaffName ,PhoneNo, StaffEmail, DepartmentID)

Auditor Table:

FD1: StaffID → StaffName, PhoneNo, StaffEmail

- **Step 1**: Primary Key? Yes, StaffID
    - **1NF Complete**
- **Step 2:** 2NF Partial-key dependencies? No
    - **2NF Complete**
- **Step 3:** 3NF Transitive dependencies? No
    - **3NF Complete**
- **Step 4**: BCNF Non-candidate Keys? No
    - **BCNF Complete**
- **Relations:**
    - **R1**: (StaffID, StaffName, PhoneNo, StaffEmail)

Employee Table:

FD1: StaffID → StaffName, PhoneNo, StaffEmail

- **Step 1:** Primary Key? Yes, StaffID
    - **1NF Complete**
- **Step 2**: 2NF Partial-key dependencies? No
    - **2NF Complete**
- S**tep 3**: 3NF Transitive dependencies? No

    - **3NF Complete**
- **Step 4:** BCNF Non-candidate Keys? No
    - **BCNF Complete**
- Relations:
    - **R1:**(StaffID, StaffName, PhoneNo, StaffEmail)

Travel_Request Table:

FD1: TravelRequestID → Destination, DepartureDate, ReturnDate, EstimatedCost, SubmissionDate, StaffID

- **Step 1**: Primary Key? Yes, TravelRequestID
    - **1NF Complete**
- **Step 2:** 2NF Partial-key dependencies? No
    - **2NF Complete**
- **Step 3:** 3NF Transitive dependencies? No
    - **3NF Complete**
- **Step 4:** BCNF Non-candidate Keys? No
    - **BCNF Complete**
- **Relations:**
    - **R1**: (TravelRequestID, Destination, DepartureDate, ReturnDate, EstimatedCost, SubmissionDate, StaffID)

Reimbursement_Request Table:

FD1: ReimbursementID → Amount, ReimbursementRequestDate, AmountPaid, PaymentStatus, TravelRequestID, StaffID
FD2: TravelRequestID → StaffID

- **Step 1:** Primary Key? ReimbursementID
    - **1NF Complete**
- **Step 2:** 2NF Partial-key dependencies? No
    - **2NF Complete**
- **Step 3:** 3NF Transitive dependencies? Yes
    - FD1: (ReimbursementID , Amount, ReimbursementRequestDate, AmountPaid, PaymentStatus, TravelRequestID, StaffID)
    - FD2: (TravelRequestID ,StaffID)
    - Splitting :
        - R1: (ReimbursementID , Amount, ReimbursementRequestDate, AmountPaid, PaymentStatus, TravelRequestID)
        - R2: (TravelRequestID , StaffID)
        - **3NF complete**
- **Step 4:** BCNF Non-candidate Keys? No
    - **BCNF Complete**
- **Relations:**

- **R1:** (ReimbursementID, Amount, ReimbursementRequestDate, AmountPaid, PaymentStatus, TravelRequestID)
- **R2:** (TravelRequestID, StaffID)

Approval_Track Table:

FD1: ApprovalID → ApprovalStatus, ApprovalDate, ApprovalComments, StaffID (fk)

- **Step 1:** Primary Key? ApprovalID
  - **1NF Complete**
- **Step 2:** 2NF Partial-key dependencies? No
  - **2NF Complete**
- **Step 3:** 3NF Transitive dependencies? No
  - **3NF Complete**
- **Step 4:** BCNF Non-candidate Keys? No
  - **BCNF Complete**
- **Relation:**
  - **R1**: (ApprovalID, ApprovalStatus, ApprovalDate, ApprovalComments, StaffID(fk))

TABLES AND INSERT STATEMENTS

TRAVEL_VENDOR

```
CREATE TABLE Travel_Vendor (
    VendorID VARCHAR(10) PRIMARY KEY NOT NULL,
    VendorName VARCHAR(100),
    InvoiceNo VARCHAR(50),
    VendorEmail VARCHAR(100),
    PhoneNo VARCHAR(20),
    VendorCategory VARCHAR(50)
);
```

Insert Statement

- INSERT INTO Travel_Vendor VALUES('V2000', 'SkyHigh Airlines', 'INV-3822', 'info@skyhighair.com', '(245)473-3695', 'Airline')
- INSERT INTO Travel_Vendor VALUES ('V2001', 'GrandView Hotels', 'INV-6747', 'reservations@grandview.com', '(261)842-2902', 'Hotel')
- INSERT INTO Travel_Vendor VALUES('V2002', 'GoCar Rentals', 'INV-3886', 'service@gocar.com', '(174)805-3602', 'Car')
- INSERT INTO Travel_Vendor VALUES('V2003', 'TravelSecure', 'INV-8505', 'support@travelsecure.com', '(592)069-1190', 'Other'),
- INSERT INTO Travel_Vendor VALUES('V2004', 'JetLink Airlines', 'INV-9003', 'bookings@jetlink.com', '(822)796-1756', 'Airline'),

- INSERT INTO Travel_Vendor VALUES ('V2005', 'Lux Corporate Suites', 'INV-4889', 'lux@corporatesuites.com', '(800) 567-8901', 'Rental')
- INSERT INTO Travel_Vendor VALUES('V2006', 'World Car Service', 'INV-5167', 'schedule@worldamcar.com', '(555) 891-1023', 'Taxi')
- INSERT INTO Travel_Vendor VALUES('V2007', 'GreatView Air', 'INV-7486', 'booking@greatviewair.com', '(555) 842-2902', 'Airline')

VendorID	VendorNam	InvoiceNo	VendorEmai	PhoneNo	VendorCate	Click to Add
V2000	SkyHigh Airline	INV-3822	info@skyhigha	(245)473-3695	Airline	
V2001	GrandView Hot	INV-6747	reservations@	(261)842-2902	Hotel	
V2002	GoCar Rentals	INV-3886	service@gocar	(174)805-3602	Car	
V2003	TravelSecure	INV-8505	support@trave	(592)069-1190	Other	
V2004	JetLink Airlines	INV-9003	bookings@jetl	(822)796-1756	Airline	
V2005	Lux Corporate S	INV-4889	lux@corporate	(800)567-8901	Rental	
V2006	World Car Serv	INV-5167	schedule@wor	(555)891-1023	Taxi	
V2007	GreatView Air	INV-7486	booking@grea	(555)842-2902	Airline	

STAFF

```
CREATE TABLE Staff(
    StaffID VARCHAR (5) NOT NULL,
    StaffName VARCHAR(50) NOT NULL,
    PhoneNo VARCHAR (15) NOT NULL,
    StaffEmail VARCHAR(50),
    DepartmentID NUMBER NOT NULL
);
```

Primary Key

```
ALTER TABLE Staff
ADD CONSTRAINT pk_Staff PRIMARY KEY (StaffID)
```

Foreign Key

```
ALTER TABLE Staff
ADD CONSTRAINT fk_Staff FOREIGN KEY (DepartmentID) REFERENCES Department
(DepartmentID)
```

Insert Statement

- INSERT INTO Staff VALUES ('E3441', 'Jeremy Young', '(555) 301-8871', 'jeremy.young@bearcatwealth.com', 1001)
- INSERT INTO Staff VALUES ('E3442', 'Kaylah Moon', '(555) 288-4359', 'kaylah.moon@bearcatwealth.com', 1001)
- INSERT INTO Staff VALUES ('E3443', 'Kristen Gomez', '(555) 419-7202', 'kristen.gomez@bearcatwealth.com', 1002)

- INSERT INTO Staff VALUES ('E3444', 'Frank Rodgers', '(555) 376-5408', 'frank.rodgers@bearcatwealth.com', 1001)
- INSERT INTO Staff VALUES ('E3445', 'Reid Stein', '(555) 230-1914', 'reid.stein@bearcatwealth.com', 1005)
- INSERT INTO Staff VALUES ('E3446', 'Hector Dalton', '(555) 918-6033', 'hector.dalton@bearcatwealth.com', 1004)
- INSERT INTO Staff VALUES ('E3447', 'Mariela Schultz', '(555) 822-7709', 'mariela.schultz@bearcatwealth.com', 1002)
- INSERT INTO Staff VALUES ('E3448', 'Angela Molina', '(555) 374-2086', 'angela.molina@bearcatwealth.com', 1003)
- INSERT INTO Staff VALUES ('E3449', 'Gerald Preston', '(555) 648-4157', 'gerald.preston@bearcatwealth.com', 1004)
- INSERT INTO Staff VALUES ('E3450', 'Reilly Moyer', '(555) 737-9440', 'reilly.moyer@bearcatwealth.com', 1001)
- INSERT INTO Staff VALUES ('E3451', 'Carlee Yu', '(555) 263-1085', 'carlee.yu@bearcatwealth.com', 1005)

StaffID	StaffName	PhoneNo	StaffEmail	Department
E3441	Jeremy Young	(555) 301-8871	jeremy.young@bearcatwealth.com	1001
E3442	Kaylah Moon	(555) 288-4359	kaylah.moon@bearcatwealth.com	1001
E3443	Kristen Gomez	(555) 419-7202	kristen.gomez@bearcatwealth.com	1002
E3444	Frank Rodgers	(555) 376-5408	frank.rodgers@bearcatwealth.com	1001
E3445	Reid Stein	(555) 230-1914	reid.stein@bearcatwealth.com	1005
E3446	Hector Dalton	(555) 918-6033	hector.dalton@bearcatwealth.com	1004
E3447	Mariela Schultz	(555) 822-7709	mariela.schultz@bearcatwealth.com	1002
E3448	Angela Molina	(555) 374-2086	angela.molina@bearcatwealth.com	1003
E3449	Gerald Preston	(555) 648-4157	gerald.preston@bearcatwealth.com	1004
E3450	Reilly Moyer	(555) 737-9440	reilly.moyer@bearcatwealth.com	1001
E3451	Carlee Yu	(555) 263-1085	carlee.yu@bearcatwealth.com	1005

EMPLOYEE

```
CREATE TABLE Employee(
        StaffID VARCHAR (20) NOT NULL,
        StaffName VARCHAR(50),
        PhoneNo VARCHAR(50),
        StaffEmail VARCHAR(50)
)
```

Primary Key

```
ALTER TABLE Employee
ADD CONSTRAINT pk_Employee PRIMARY  KEY (StaffID) ;
```

Foreign Key

ALTER TABLE Employee
ADD CONSTRAINT fk_Employee FOREIGN KEY (StaffID) REFERENCES Staff (StaffID);

Insert Statement:

- INSERT INTO Employee VALUES ('E3445', 'Reid Stein', '(555) 230-1914', 'reid.stein@bearcatwealth.com')
- INSERT INTO Employee VALUES ('E3446', 'Hector Dalton', '(555) 918-6033', 'hector.dalton@bearcatwealth.com')
- INSERT INTO VALUES ('E3447', 'Mariela Schultz', '(555) 822-7709', 'mariela.schultz@bearcatwealth.com')
- INSERT INTO VALUES ('E3448', 'Angela Molina', '(555) 374-2086', 'angela.molina@bearcatwealth.com')
- INSERT INTO VALUES ('E3449', 'Gerald Preston', '(555) 648-4157', 'gerald.preston@bearcatwealth.com')
- INSERT INTO Employee VALUES ('E3451', 'Carlee Yu', '(555) 263-1085', 'carlee.yu@bearcatwealth.com')
- INSERT INTO Employee VALUES ('E3443', 'Kristen Gomez', '(555) 419-7202', 'kristen.gomez@bearcatwealth.com')

StaffID	StaffName	PhoneNo	StaffEmail
E3443	Kristen Gomez	(555) 419-7202	kristen.gomez@bearcatwealth.com
E3445	Reid Stein	(555) 230-1914	reid.stein@bearcatwealth.com
E3446	Hector Dalton	(555) 918-6033	hector.dalton@bearcatwealth.com
E3447	Mariela Schultz	(555) 822-7709	mariela.schultz@bearcatwealth.com
E3448	Angela Molina	(555) 374-2086	angela.molina@bearcatwealth.com
E3449	Gerald Preston	(555) 648-4157	gerald.preston@bearcatwealth.com
E3451	Carlee Yu	(555) 263-1085	carlee.yu@bearcatwealth.com

AUDITOR

```
CREATE TABLE Auditor(
        StaffID VARCHAR (20) NOT NULL,
        StaffName VARCHAR(50),
        PhoneNo VARCHAR(50),
        StaffEmail VARCHAR(50)
)
```

Primary Key

ALTER TABLE Auditor
ADD CONSTRAINT pk_Employee PRIMARY KEY (StaffID);

Foreign Key

ALTER TABLE Auditor
ADD CONSTRAINT fk_Auditor FOREIGN KEY (StaffID) REFERENCES Staff (StaffID);

Insert Statement:

- INSERT INTO Auditor VALUES ('E3441', 'Jeremy Young', '(555) 301-8871', 'jeremy.young@bearcatwealth.com')
- INSERT INTO Auditor VALUES ('E3450', 'Reilly Moyer', '(555) 737-9440', 'reilly.moyer@bearcatwealth.com')
- INSERT INTO Auditor VALUES ('E3442', 'Kaylah Moon', '(555) 288-4359', 'kaylah.moon@bearcatwealth.com')
- INSERT INTO Auditor VALUES ('E3444', 'Frank Rodgers', '(555) 376-5408', 'frank.rodgers@bearcatwealth.com')

StaffID	StaffName	PhoneNo	StaffEmail
E3441	Jeremy Young	(555) 301-8871	jeremy.young@bearcatwealth.com
E3442	Kaylah Moon	(555) 288-4359	kaylah.moon@bearcatwealth.com
E3444	Frank Rodgers	(555) 376-5408	frank.rodgers@bearcatwealth.com
E3450	Reilly Moyer	(555) 737-9440	reilly.moyer@bearcatwealth.com

TRAVEL_REQUEST

```
CREATE TABLE Travel_Request(
        TravelRequestID VARCHAR (6) NOT NULL,
        Destination VARCHAR (50),
        DepartureDate DATE,
        ReturnDate DATE,
        EstimatedCost NUMBER NOT NULL,
        SubmissionDate DATE,
        StaffID VARCHAR(5) NOT NULL
)
```

Primary Key

ALTER TABLE Travel_Request
ADD CONSTRAINT pk_Travel_Request PRIMARY KEY (TravelRequestID)

Foreign Key

ALTER TABLE Travel_Request
ADD CONSTRAINT fk_Travel_Request FOREIGN KEY (StaffID) REFERENCES Staff (StaffID)

Insert Statement

- INSERT INTO Travel_Request VALUES ('TR4000', 'San Francisco, California', '03/17/2025', '03/21/2025', 1793.00, '03/12/2025', 'E3446')
- INSERT INTO Travel_Request VALUES ('TR4001', 'Paris, France', '04/06/2025', '04/09/2025', 2303.00, '04/01/2025', 'E3445')
- INSERT INTO Travel_Request VALUES ('TR4002', 'Hong Kong, China', '03/27/2025', '04/03/2025', 2927.00, '03/22/2025', 'E3447')
- INSERT INTO Travel_Request VALUES ('TR4003', 'London, England', '03/07/2025', '03/09/2025', 2679.00, '03/02/2025', 'E3443')
- INSERT INTO Travel_Request VALUES ('TR4004', 'Chicago, Illinois', '03/06/2025', '03/11/2025', 1397.00, '03/01/2025', 'E3447')
- INSERT INTO Travel_Request VALUES ('TR4005', 'Tokyo, Japan', '04/08/2025', '04/14/2025', 2544.00, '04/03/2025', 'E3448')
- INSERT INTO Travel_Request VALUES ('TR4006', 'New York, New York', '03/24/2025', '03/26/2025', 1455.00, '03/19/2025', 'E3449')
- INSERT INTO Travel_Request VALUES ('TR4007', 'Austin, Texas', '03/23/2025', '03/26/2025', 992.00, '03/18/2025', 'E3450')

TravelReque	Destination	DepartureDate	ReturnDate	EstimatedCost	SubmissionI	StaffI
TR4000	San Francisco, California	3/17/2025	3/21/2025	1793	3/12/2025	E3446
TR4001	Paris, France	4/6/2025	4/9/2025	2303	4/1/2025	E3445
TR4002	Hong Kong, China	3/27/2025	4/3/2025	2927	3/22/2025	E3447
TR4003	London, England	3/7/2025	3/9/2025	2679	3/2/2025	E3443
TR4004	Chicago, Illinois	3/6/2025	3/11/2025	1397	3/1/2025	E3447
TR4005	Tokyo, Japan	4/8/2025	4/14/2025	2544	4/3/2025	E3448
TR4006	New York, New York	3/24/2025	3/26/2025	1455	3/19/2025	E3449
TR4007	Austin, Texas	3/23/2025	3/26/2025	992	3/18/2025	E3450

EXPENSE_RECORD

```
CREATE TABLE Expense_Record (
        ExpenseID VARCHAR(10) NOT NULL,
        ExpenseDate DATE,
        ExpenseAmt DECIMAL(10, 2),
        ExpenseCategory VARCHAR(50),
        VendorID VARCHAR(10),
        TravelRequestID VARCHAR(10)
)
```

Primary Key

```
ALTER TABLE Expense_Record
ADD CONSTRAINT pk_expense_record PRIMARY KEY (ExpenseID)
```

Foreign Key

ALTER TABLE Expense_Record
ADD CONSTRAINT fk_Exp_Ven FOREIGN KEY (VendorID) REFERENCES Travel_Vendor(VendorID)

ALTER TABLE Expense_Record
ADD CONSTRAINT fk_Expense_Rec_Travel_Req FOREIGN KEY (TravelRequestID) REFERENCES Travel_Request(TravelRequestID)

ALTER TABLE Expense_Record
ADD CONSTRAINT fk_Exp_ExpCat FOREIGN KEY (ExpenseCategory) REFERENCES ExpenseCategory(ExpenseCategory);

Insert Statement

- INSERT INTO Expense_Record VALUES ('X3000', #03/19/2025#, 1750.00, 'Flight', 'V2000', 'TR4000')
- INSERT INTO Expense_Record VALUES ('X3001', #04/08/2025#, 330.00, 'Taxi', 'V2006', 'TR4001');
- INSERT INTO Expense_Record VALUES ('X3002', #03/29/2025#, 881.00, 'Flight', 'V2004', 'TR4002')
- INSERT INTO Expense_Record VALUES ('X3003', #03/08/2025#, 1351.00, 'Taxi', 'V2002', 'TR4003')
- INSERT INTO Expense_Record VALUES ('X3004', #03/09/2025#, 745.00, 'Meals', 'V2003', 'TR4004')
- INSERT INTO Expense_Record VALUES ('X3005', #04/12/2025#, 1481.00, 'Hotel', 'V2001', 'TR4005')
- INSERT INTO Expense_Record VALUES ('X3006', #03/24/2025#, 806.00, 'Flight', 'V2007', 'TR4006')
- INSERT INTO Expense_Record VALUES ('X3007', #03/23/2025#, 1025.00, 'Flight', 'V2000', 'TR4007')

ExpenseID	ExpenseDate	ExpenseAmt	ExpenseCateg	VendorID	TravelRequestID
X3000	3/19/2025	1750	Flight	V2000	TR4000
X3001	4/8/2025	330	Taxi	V2006	TR4001
X3002	3/29/2025	881	Flight	V2004	TR4002
X3003	3/8/2025	1351	Taxi	V2002	TR4003
X3004	3/9/2025	745	Meals	V2003	TR4004
X3005	4/12/2025	1481	Hotel	V2001	TR4005
X3006	3/24/2025	806	Flight	V2007	TR4006
X3007	3/23/2025	1025	Flight	V2000	TR4007

EXPENSE_CATEGORY

```
CREATE TABLE ExpenseCategory (
        ExpenseCategory VARCHAR(50) NOT NULL,
        CategoryMaxLimit DECIMAL(10, 2),
)
```

Primary Key

```
ALTER TABLE ExpenseCategory
ADD CONSTRAINT pk_expensecategory PRIMARY KEY (ExpenseCategory);
```

Insert Statement

- INSERT INTO ExpenseCategory VALUES ('Flight', 1500)
- INSERT INTO ExpenseCategory VALUES ('Hotel', 2500)
- INSERT INTO ExpenseCategory VALUES ('Meals', 600)
- INSERT INTO ExpenseCategory VALUES ('Taxi', 300)
- INSERT INTO ExpenseCategory VALUES ('Other', 400)

ExpenseCategory	CategoryMa
Flight	1500
Hotel	2500
Meals	600
Taxi	300
Other	400

RECEIPTS

```
CREATE TABLE Receipts(
        ReceiptNo VARCHAR(50) NOT NULL,
        UploadDate DATE,
        UploadFileNo VARCHAR(50) NOT NULL,
        TotalAmt DECIMAL,
        ExpenseID VARCHAR(50) NOT NULL
)
```

Primary Key

```
ALTER TABLE Receipts
ADD CONSTRAINT pk_receipts PRIMARY KEY (ReceiptNo)
```

Foreign Key

```
ALTER TABLE Receipts
ADD CONSTRAINT fk_receipts FOREIGN KEY (ExpenseID) REFERENCES Expense_Record(ExpenseID)
```

Insert Statement

- INSERT INTO Receipts VALUES('R5000','3/20/2025','UPL-32872',525.00,'X3000')
- INSERT INTO Receipts VALUES('R5001','4/9/2025','UPL-33855',430.00,'X3001')
- INSERT INTO Receipts VALUES('R5002','3/30/2025','UPL-22634',881.00,'X3002')
- INSERT INTO Receipts VALUES('R5003','3/9/2025','UPL-36624',1351.00,'X3003')
- INSERT INTO Receipts VALUES('R5004','3/10/2025','UPL-58524',745.00,'X3004')

ReceiptNo	UploadDate	UploadFileN	TotalAmt	ExpenseID
R5000	3/20/2025	UPL-32872	$525.00	X3000
R5001	4/9/2025	UPL-33855	$430.00	X3001
R5002	3/30/2025	UPL-22634	$881.00	X3002
R5003	3/9/2025	UPL-36624	$1,351.00	X3003
R5004	3/10/2025	UPL-58524	$745.00	X3004

PAYMENT_RECORD

```
CREATE TABLE Payment_Record(
        PaymentNo VARCHAR(50) NOT NULL,
        Amount DECIMAL(10,2),
        PaymentRecordDate DATE,
        PaymentType VARCHAR(50),
        ExpenseID VARCHAR(50)
)
```

Primary Key

```
ALTER TABLE Payment_Record
ADD CONSTRAINT pk_Payment_Record PRIMARY KEY (PaymentNo)
```

Foreign_Key

```
ALTER TABLE Payment_Record
ADD CONSTRAINT fk_Payment_Record FOREIGN KEY (ExpenseID)
REFERENCES Expense_Record(ExpenseID)
```

Insert Statement

- INSERT INTO Payment_Record VALUES('P6000', 525.00, '3/21/2025', 'Bank Transfer', 'X3000')
- INSERT INTO Payment_Record VALUES('P6001', 430.00, '4/10/2025', 'Credit Card', 'X3001')
- INSERT INTO Payment_Record VALUES('P6002', 881.00, '3/31/2025', 'Bank Transfer', 'X3002')
- INSERT INTO Payment_Record VALUES('P6003', 1351.00, '3/10/2025', 'Credit Card', 'X3003')
- INSERT INTO Payment_Record VALUES('P6004', 745.00, '3/11/2025', 'Bank Transfer', 'X3004')

PaymentNo	Amount	PaymentRec	PaymentTyp	ExpenseID
P6000	$525.00	3/21/2025	Bank Transfer	X3000
P6001	$430.00	4/10/2025	Credit Card	X3001
P6002	$881.00	3/31/2025	Bank Transfer	X3002
P6003	$1,351.00	3/10/2025	Credit Card	X3003
P6004	$745.00	3/11/2025	Bank Transfer	X3004

DEPARTMENT

CREATE TABLE Department (
DepartmentID VARCHAR(20) NOT NULL,
DeptName VARCHAR(20),
Location VARCHAR(20)
)

Primary Key

ALTER TABLE Department
ADD CONSTRAINT pk_Department PRIMARY KEY(DepartmentID)

Insert Statement

- INSERT INTO Department VALUES ('1001', 'Accounting', 'New York')
- INSERT INTO Department VALUES ('1002', 'Marketing', 'Chicago')
- INSERT INTO Department VALUES ('1003', 'Sales', 'San Francisco')
- INSERT INTO Department VALUES ('1004', 'Finance', 'Boston')
- INSERT INTO Department VALUES ('1005', 'HR', 'Seattle')

DepartmentI	DeptName	Location
1001	Accounting	New York
1002	Marketing	Chicago
1003	Sales	San Francisco
1004	Finance	Boston
1005	HR	Seattle

REIMBURSEMENT_REQUEST

CREATE TABLE Reimbursement_Request (
ReimbursementID VARCHAR(20) PRIMARY KEY,
Amount DECIMAL(10, 2) NOT NULL,
ReimbursementRequestDate DATE NOT NULL,
AmountPaid DECIMAL(10, 2),
PaymentStatus VARCHAR(20),
TravelRequestID VARCHAR(20),
StaffID VARCHAR(20)
)

Foreign Key

ALTER TABLE Reimbursement_Request
ADD CONSTRAINT fk_RR_ST FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)

ALTER TABLE Reimbursement_Request
ADD CONSTRAINT fk_RR_TR FOREIGN KEY (TravelRequestID) REFERENCES
Travel_Request(TravelRequestID)

Insert statement

- INSERT INTO Reimbursement_Request VALUES('RE7000', 1793.00, '2025-03-18', 682.00, 'Paid', 'TR4000', 'E3446')
- INSERT INTO Reimbursement_Request VALUES(('RE7001', 2303.00, '2025-04-07', 1305.00, 'Rejected', 'TR4001', 'E3445')
- INSERT INTO Reimbursement_Request VALUES(('RE7002', 2927.00, '2025-03-28', 275.00, 'Paid', 'TR4002', 'E3447')
- INSERT INTO Reimbursement_Request VALUES(('RE7003', 2679.00, '2025-03-08', 532.00, 'Pending', 'TR4003', 'E3443')
- INSERT INTO Reimbursement_Request VALUES(('RE7004', 1397.00, '2025-03-07', 487.00, 'Paid', 'TR4004', 'E3447')
- INSERT INTO Reimbursement_Request VALUES(('RE7005', 2544.00, '2025-04-09', 532.00, 'Pending', 'TR4005', 'E3448')
- INSERT INTO Reimbursement_Request VALUES(('RE7006', 1455.00, '2025-03-25', 446.00, 'Paid', 'TR4006', 'E3449')
- INSERT INTO Reimbursement_Request VALUES(('RE7007', 992.00, '2025-03-24', 587.00, 'Rejected', 'TR4007', 'E3450');

Reimbursem	Amount	Reimbursem	AmountPaid	PaymentStat	TravelReque	StaffID
RE7000	1793	18/03/2025	682	Paid	TR4000	E3446
RE7001	2303	07/04/2025	1305	Rejected	TR4001	E3445
RE7002	2927	28/03/2025	275	Paid	TR4002	E3447
RE7003	2679	08/03/2025	532	Pending	TR4003	E3443
RE7004	1397	07/03/2025	487	Paid	TR4004	E3447
RE7005	2544	09/04/2025	532	Pending	TR4005	E3448
RE7006	1455	25/03/2025	446	Paid	TR4006	E3449
RE7007	992	24/03/2025	587	Rejected	TR4007	E3450

APPROVAL_TRACK

```
CREATE TABLE Approval_Track (
        ApprovalID VARCHAR(20) NOT NULL,
        ApprovalStatus VARCHAR(20),
        ApprovalDate DATE,
        ApprovalComments VARCHAR(100),
        StaffID VARCHAR(20),
)
```

Primary Key

```
ALTER TABLE Approval_Track
ADD CONSTRAINT pk_Approval_Track PRIMARY KEY(ApprovalID)
```

Foreign Key

```
ALTER TABLE Approval_Track
ADD CONSTRAINT fk_Travel_Request_StaffID  FOREIGN KEY (StaffID)REFERENCES Staff (StaffID)
```

Insert Statement

- INSERT INTO Approval_Track VALUES ('A8000', 'Approved', '4/26/2025','Let finance know if plans change.', 'E3442')
- INSERT INTO Approval_Track VALUES ('A8001', 'Pending', '5/3/2025','Identify key meeting dates.', 'E3450')
- INSERT INTO Approval_Track VALUES ('A8002', 'Pending', '4/21/2025','Waiting for new estimates.', 'E3441')
- INSERT INTO Approval_Track VALUES ('A8003', 'Rejected', '5/2/2025',' Please submit receipts.', 'E3450')
- INSERT INTO Approval_Track VALUES ('A8004', 'Approved', '4/21/2025','Thank you for planning ahead.', 'E3441')
- INSERT INTO Approval_Track VALUES ('A8005', 'Approved', '4/28/2025',' Safe travels and please follow policy.', 'E3445')
- INSERT INTO Approval_Track VALUES ('A8006', 'Approved', '4/21/2025',' Coordinate with your team.', 'E3445')
- INSERT INTO Approval_Track VALUES ('A8007', 'Pending', '4/29/2025',' Staying within budget is a concern.', 'E3451')

ApprovalID	ApprovalStat	ApprovalDat	ApprovalComments	StaffID	Cli
A8000	Approved	26/04/2025	Let finance know if plans change.	E3442	
A8001	Pending	03/05/2025	Identify key meeting dates.	E3450	
A8002	Pending	21/04/2025	Waiting for new estimates.	E3441	
A8003	Rejected	02/05/2025	Please submit receipts.	E3450	
A8004	Approved	21/04/2025	Thank you for planning ahead.	E3441	
A8005	Approved	28/04/2025	Safe travels and please follow policy.	E3445	
A8006	Approved	21/04/2025	Coordinate with your team.	E3445	
A8007	Pending	29/04/2025	Staying within budget is a concern.	E3451	

SCENARIOS

Scenario 1: a) Identify vendors who have been paid and have valid receipt records to ensure accurate financial tracking and policy compliance.

SELECT VendorName, VendorCategory, InvoiceNo, Amount AS PaidAmount, TotalAmt As ReceiptTotal
FROM Travel_Vendor v, Expense_Record er, Payment_Record pr, Receipts r
WHERE v.VendorID = er.VendorID AND er.ExpenseID = pr.ExpenseID AND er.ExpenseID = r.ExpenseID

VendorName	VendorCate	InvoiceNo	PaidAmount	ReceiptTota
SkyHigh Airlines	Airline	INV-3822	$525.00	$525.00
World Car Service	Taxi	INV-5167	$430.00	$430.00
JetLink Airlines	Airline	INV-9003	$881.00	$881.00
GoCar Rentals	Car	INV-3886	$1,351.00	$1,351.00
TravelSecure	Other	INV-8505	$745.00	$745.00

b) Identify vendors with recorded expenses that have not yet been paid, to track pending payments and manage outstanding liabilities.

```
SELECT  VendorName, InvoiceNo, er.ExpenseAmt
FROM (Travel_Vendor  v INNER JOIN Expense_Record  er ON v.VendorID = er.VendorID)
LEFT OUTER JOIN Payment_Record  pr ON er.ExpenseID = pr.ExpenseID
WHERE pr.PaymentNo IS NULL
```

VendorName	InvoiceNo	ExpenseAmt
GrandView Hotels	INV-6747	1481.00
GreatView Air	INV-7486	806.00
SkyHigh Airlines	INV-3822	1025.00

Scenario 2: **Identify expenses that exceed their category's maximum allowed amount to detect policy violations and prevent overspending.**

```
SELECT e.ExpenseID, e.ExpenseAmt, c.CategoryMaxLimit
FROM Expense_Record e
INNER JOIN ExpenseCategory c ON e.ExpenseCategory = c.ExpenseCategory
WHERE e.ExpenseAmt > (
  SELECT CategoryMaxLimit
  FROM ExpenseCategory
  WHERE ExpenseCategory = e.ExpenseCategory
);
```

ExpenseID	ExpenseAmt	CategoryMa
X3000	1750.00	$1,500.00
X3001	330.00	$300.00
X3002	881.00	$1,500.00
X3004	745.00	$600.00
X3006	806.00	$1,500.00

Scenario 3: **List the names of employees, the travel request ID, and total expense amount only for those travel requests whose total expenses exceed $800.**

```
SELECT e.StaffName, tr.TravelRequestID, (SELECT SUM(er.ExpenseAmt)
FROM Expense_Record AS er
WHERE er.TravelRequestID = tr.TravelRequestID) AS TotalExpense
FROM Employee AS e
INNER JOIN Travel_Request AS tr ON e.StaffID = tr.StaffID
WHERE (SELECT SUM(er.ExpenseAmt)
FROM Expense_Record AS er
WHERE er.TravelRequestID = tr.TravelRequestID) > 800
```

StaffName	TravelRequestID	TotalExpense
Kristen Gomez	TR4003	1351
Hector Dalton	TR4000	1750
Mariela Schultz	TR4002	881
Angela Molina	TR4005	1481
Gerald Preston	TR4006	806

Scenario 4: Identify all Departments located in cities that start with "S."

```
SELECT *
FROM Department
WHERE location LIKE 'S*'
```

DepartmentI	DeptName	Location
1003	Sales	San Francisco
1005	HR	Seattle

Scenario 5: List employees who have submitted more than one travel request to monitor frequent travel and optimize resource allocation.

```
SELECT StaffName, COUNT(TravelRequestID) AS NumberOfRequests
FROM Staff s INNER JOIN Travel_Request tr
ON s.StaffID = tr.StaffID
GROUP BY StaffName
HAVING COUNT(TravelRequestID) > 1;
```

StaffName	NumberOfRequests
Mariela Schultz	2

Scenario 6: Show the number of days it took to upload digital copies of receipts in the system.

```
SELECT  e.ExpenseID,  e.ExpenseDate, r.UploadDate,  DateDiff("d", e.ExpenseDate, r.UploadDate)
AS DaysToUpload
FROM Expense_Record AS e
INNER JOIN Receipts AS r ON e.ExpenseID = r.ExpenseID
ORDER BY DateDiff("d", e.ExpenseDate, r.UploadDate) DESC;
```

ExpenseID	ExpenseDat	UploadDate	DaysToUplo
X3004	3/9/2025	3/10/2025	1
X3003	3/8/2025	3/9/2025	1
X3002	3/29/2025	3/30/2025	1
X3001	4/8/2025	4/9/2025	1
X3000	3/19/2025	3/20/2025	1
*			

Scenario 7: Find all payments made by a credit card; this can help auditors find out how many times the employees use their company card.

```
SELECT * FROM Payment_Record
WHERE PaymentType = "Credit Card";
```

PaymentNo	Amount	PaymentRec	PaymentTyp	ExpenseID
P6001	$430.00	4/10/2025	Credit Card	X3001
P6003	$1,351.00	3/10/2025	Credit Card	X3003

Scenario 8: Identify the highest receipt amount to flag for audit review, ensuring spending stays within policy limits.

```
SELECT TOP 1
FORMAT(TotalAmt, '$#,##0.00') AS MaxReceiptAmount, ReceiptNo, ExpenseID
FROM Receipts
ORDER BY TotalAmt DESC;
```

MaxReceipt	ReceiptNo	ExpenseID
$1,351.00	R5003	X3003

Scenario 9: Identify the highest receipt amount to flag for audit review, ensuring spending stays within policy limits.

```
SELECT e.StaffName, 'No Expenses' AS ExpenseStatus
FROM Employee e
WHERE e.StaffID NOT IN (SELECT tr.StaffID FROM Travel_Request tr INNER JOIN Expense_Record
er ON tr.TravelRequestID = er.TravelRequestID)
UNION
SELECT e.StaffName, 'Has Expenses' AS ExpenseStatus
FROM Employee e
WHERE e.StaffID IN (SELECT tr.StaffID FROM Travel_Request tr INNER JOIN Expense_Record er ON
tr.TravelRequestID = er.TravelRequestID)
```

StaffName	ExpenseStatus
Angela Molina	Has Expenses
Carlee Yu	No Expenses
Gerald Preston	Has Expenses
Hector Dalton	Has Expenses
Kristen Gomez	Has Expenses
Mariela Schultz	Has Expenses
Reid Stein	Has Expenses

Scenario 10: List all vendors in the Flight, Hotel, or Transportation categories along with their contact emails for communication and record-keeping.

```
SELECT VendorName, VendorCategory, VendorEmail
FROM Travel_Vendor
WHERE VendorCategory
IN ('Flight', 'Hotel', 'Transportation')
```

VendorName	VendorCate	VendorEmail
GrandView Hotels	Hotel	reservations@grandview.com

Scenario 11: List of the top 3 employees who submitted travel requests with the highest total estimated costs. Include the employee's name, Department, and their total estimated travel costs.

```
SELECT TOP 3  s.StaffName, d.DeptName, SUM(tr.EstimatedCost) AS TotalEstimatedCost
FROM (Travel_Request AS tr INNER JOIN Staff AS s ON tr.StaffID = s.StaffID)
INNER JOIN Department AS d ON s.DepartmentID = d.DepartmentID
GROUP BY s.StaffName, d.DeptName
ORDER BY SUM(tr.EstimatedCost) DESC
```

StaffName	DeptName	TotalEstimatedCost
Mariela Schultz	Marketing	4324
Kristen Gomez	Marketing	2679
Angela Molina	Sales	2544

Scenario 12: Retrieve all approval requests submitted after April 21, 2025, ordered by the approval date.

```
SELECT *
FROM Approval_Track
WHERE ApprovalDate > #04/21/2025#
ORDER BY ApprovalDate;
```

ApprovalID	ApprovalSta	ApprovalDat	ApprovalComments	StaffID
A8000	Approved	4/26/2025	Let finance know if plans change.	E3442
A8005	Approved	4/28/2025	Safe travels and please follow policy.	E3445
A8007	Pending	4/29/2025	Staying within budget is a concern.	E3451
A8003	Rejected	5/2/2025	Please submit receipts.	E3450
A8001	Pending	5/3/2025	Identify key meeting dates.	E3450

Scenario 13: Find the earliest and latest payment dates recorded in the system to analyze the payment timeline.

```
SELECT MIN(PaymentRecordDate) AS FirstPayment, MAX(PaymentRecordDate) AS LastPayment
FROM Payment_Record;
```

FirstPayment	LastPayment
3/10/2025	4/14/2025

CONCLUSION

One of the most challenging aspects of this project was writing queries that effectively addressed the informational needs of our database. Despite the difficulties, our team successfully created queries for various scenarios, applying a range of different SQL functions, such as TOP, FORMAT, INNER JOIN, ORDER BY, and more. Analyzing query results to ensure accuracy and relevance was crucial throughout this process. Overall, the experience of trial and error in query creation deepened our understanding of the topics we learned in class and practical application in the real world.

Creating the Entity Relationship Diagram (ERD) using UML notation also presented challenges. Our team had several meetings to fully grasp the relationships between all the system components. This step was time-intensive, requiring not only technical modeling skills but also a holistic understanding of how different parts of the system interact to form an integrated workflow in the real world. We learned that designing a database went beyond the technical work of ERDs and normalization. We had to ensure we understood the business and its problem(s), to identify what solution(s) were trying to be achieved, and to understand that designing the database was a process of problem-solving.

In comparison, the normalization phase was more straightforward. After finalizing the ERD, we systematically applied normalization principles to produce well-structured relations.

Before starting the project, we did not fully grasp how an organized system connects to all elements of a business process. Our database design now clearly links employees, travel requests, expenses, receipts, and vendors, transforming a complicated, error-prone process into a streamlined and trackable system.

If we were to repeat this project, we would seek clarification from our professor earlier in the process. Initially, we tried to interpret the assignment requirements and feedback independently, which led to some confusion. Once we eventually reached out, the professor provided prompt and clear guidance, helping us proceed forward with more confidence.

The system we designed enables BearcatWealth to effectively tackle the key challenges outlined at the start of the report. By centralizing travel and expense data, automating workflows, and enforcing policy compliance, the new system eliminates delays in reimbursements, reduces the risk of lost receipts, and improves budget tracking. This comprehensive solution provides the transparency, accuracy, and efficiency that BearcatWealth needs to manage travel and expenses at scale.