**Database Design Document: PrintDesk**

## 1. 📄 Introduction

**Project:** PrintDesk  
**Purpose:** This document outlines the structure of the database system for PrintDesk, a platform for analyzing printer usage using uploaded Excel files and visualized via Power BI.  
**Audience:** Developers, Database Administrators, QA team

## 2. 🚀 Business Requirements Summary

The system needs to: - Track which user printed how many pages - Identify the most/least used printers - Show departmental printing stats - Filter reports by date, user, department, etc.

## 3. 📉 Conceptual Design

### Main Entities:

* User
* PrintJob
* Printer
* Department
* Role

Relationships: - A user belongs to a department and has a role. - A print job is submitted by a user and uses a printer.

## 4. 🏢 Logical Design (Tables & Fields)

### Table: User

| Field | Type | Constraints |
| --- | --- | --- |
| UserID | INT | PRIMARY KEY, AUTO\_INCREMENT |
| Name | VARCHAR(100) | NOT NULL |
| Email | VARCHAR(100) | UNIQUE, NOT NULL |
| RoleID | INT | FOREIGN KEY |
| DepartmentID | INT | FOREIGN KEY |

### Table: Role

| Field | Type | Constraints |
| --- | --- | --- |
| RoleID | INT | PRIMARY KEY |
| RoleName | VARCHAR(50) | UNIQUE (Admin, Uploader, Viewer) |

### Table: Department

| Field | Type | Constraints |
| --- | --- | --- |
| DepartmentID | INT | PRIMARY KEY |
| DepartmentName | VARCHAR(100) | UNIQUE, NOT NULL |

### Table: Printer

| Field | Type | Constraints |
| --- | --- | --- |
| PrinterID | INT | PRIMARY KEY |
| Name | VARCHAR(100) | UNIQUE |
| DepartmentID | INT | FOREIGN KEY |

### Table: PrintJob

| Field | Type | Constraints |
| --- | --- | --- |
| JobID | INT | PRIMARY KEY |
| UserID | INT | FOREIGN KEY |
| PrinterID | INT | FOREIGN KEY |
| PagesPrinted | INT | NOT NULL |
| Timestamp | DATETIME | NOT NULL |

## 5. 🔒 Security Design

* Each user is assigned a role that restricts access to certain features.
* Access to data is filtered based on user role and department.

## 6. 📈 Indexing Plan

* Index on PrintJob.Timestamp for time-based filtering
* Index on Printer.Name and User.Name for search efficiency

## 7. 📊 Volume Estimates & Retention

* Expected volume: ~10,000–50,000 print jobs per month
* Retain only the last 12 months of data for analysis

## 8. 🚪 Backup & Recovery

* Daily backups (incremental)
* Weekly full backups
* Backups stored securely on separate media/cloud

## 9. 🚲 Environment & Tools

* DBMS: Microsoft SQL Server
* Integration via: ODBC with Python Flask
* Tools: SQL Server Management Studio (SSMS)

## 10. 🔄 Schema Change Management

* All schema changes reviewed and approved before deployment
* Use of migration scripts and version control (e.g., Git)

## 11. 📝 Appendix

**Sample Query:** Top 10 Most Used Printers

SELECT TOP 10 Printer.Name, COUNT(\*) AS UsageCount  
FROM PrintJob  
JOIN Printer ON PrintJob.PrinterID = Printer.PrinterID  
GROUP BY Printer.Name  
ORDER BY UsageCount DESC;

Let me know if you want a visual ER diagram or the actual SQL CREATE TABLE scripts.