INDIVIDUAL ASSIGNMENT

LEVEL 5

COMP50004
DATABASES AND DATA STRUCTURES
SENG2461

Name – Sukitha Chathuranga Hettimudalige
CB Number – CB014764

INSTRUCTION TO CANDIDATES

1. **Students are advised to underpin their answers with the use of references (cited using the Harvard Referencing Style).**

2. **Late submission will be awarded zero (0) unless extenuating circumstances (EC) are upheld.**

3. **Cases of plagiarism will be penalized.**

4. **The assignment should be submitted as a softcopy:**

   a. **The softcopy of the written assignment and source code must be uploaded to given link in the LMS.**

Contents

Introduction 3

Case Study 4

Scope - Employer & Internship Posting Module 5

Functional Requirements 6

Non-Functional Requirements 7

Entity Attributes List 8

Entity Relationships 9

Logical ERD 10

Physical ERD 11

Assumptions and Notes 12

Relational Schema 13

Data Dictionary 14

Normalization 19

Research on NoSQL 23

SQL Database Creation 27

Table Creation 28

Data Insertion 33

Query Reports 44

Transaction 52

References 53

Introduction

This report is about the database design for a centralized platform that enables university students in the Faculty of Computing to find and apply for internships, while allowing employers to post and manage internship opportunities. The database is divided into two distinct modules namely the Student & Application Management Module and the Employer & Internship Posting Module. The author of this report is responsible for the Employer & Internship Posting Module.

Case Study

After second year, many undergraduates have difficulty in finding internships. It is true that many students are unaware of internship possibilities that are out there which is a major contributor to this problem. Internships get communicated through several types of platforms, including social media, WhatsApp groups, and word-of-mouth. These unofficial methods might deceive students or result in them losing out on important opportunities.

The absence of a centralized platform where students have access to available internships in one location is the main reason behind these challenges. Students find it difficult to effectively obtain reliable and accurate information in the absence of such a system.

By offering a centralized, open, and user-friendly system that benefits employers and students, the suggested platform seeks to address these problems. It helps students make accurate choices and ensures quick access to verified internship programs. It gives employers with tools to advertise internships, handle applications effectively, and examine data to enhance the hiring process.

This platform will improve the internship experience by connecting employers and students, resulting in a more reliable and structured internship ecosystem that benefits both parties.

Scope - Employer & Internship Posting Module

The scope of this report focuses on the Employer & Internship Posting Module of the database design. Within this module, employers can post and manage internships, shortlist applicants, and generate reports for the internships that have already been posted.

Functional Requirements

1. Employer Registration & Profile Management
    a. Employers can sign up to the system by giving email and password and get registered by giving the details such as company name, Employer name, Address and Phone number.
    b. Email must be unique and should validate with “@”.
    c. Once registered employers should assign a person who oversees internships on the behalf of the company by giving his/her name, age, email and contact number and position.
    d. Employer can update the details of the person who oversees internships on the behalf of the company.
    e. Employers can update or delete the profile if needed.

2. Internship Posting & Management
    a. Employers can make new internship postings.
    b. Employers can update or delete internship postings.
    c. Ability to mark an internship as opened or closed.

3. Application shortlisting
    a. Employers can view all the applications received for internship.
    b. Employers can shortlist the candidates or reject.

4. Employer Side- Reports
    a. Ability to generate Reports with details such as number of applicants per internship during a specific period and number of positions filled during the specific period.

Non-Functional Requirements

1. Performance
    a. At least multiple users should be capable to use the system at once without encountering any performance issues.
    b. It must take roughly two seconds for the internship search results to load.

2. Scalability
    a. The design of the database should allow easy expansion (e.g.: adding new internship skills, internship categories, locations etc...

3. Security
    a. It is required to store passwords using encryption and hashing methods.
    b. Role-based access control for administrators, employers, and students.
    c. virus-inspected, secure file uploads.
    d. Protect from attacks such as CSRF, XSS, and SQL injection attacks.

4. Availability and Reliability
    a. During hours of operation, the system should be available 99 percent of the time.
    b. Daily database backup and recovery plan.

5. Data Integrity
    a. Verify that the data entered is correct (e.g., email format, file type restrictions).
    b. Avoid duplication of internships posts or accounts twice.

6. Maintainability
    a. The system should be easy to update without downtime.
    b. Code and database structure should be well-documented.

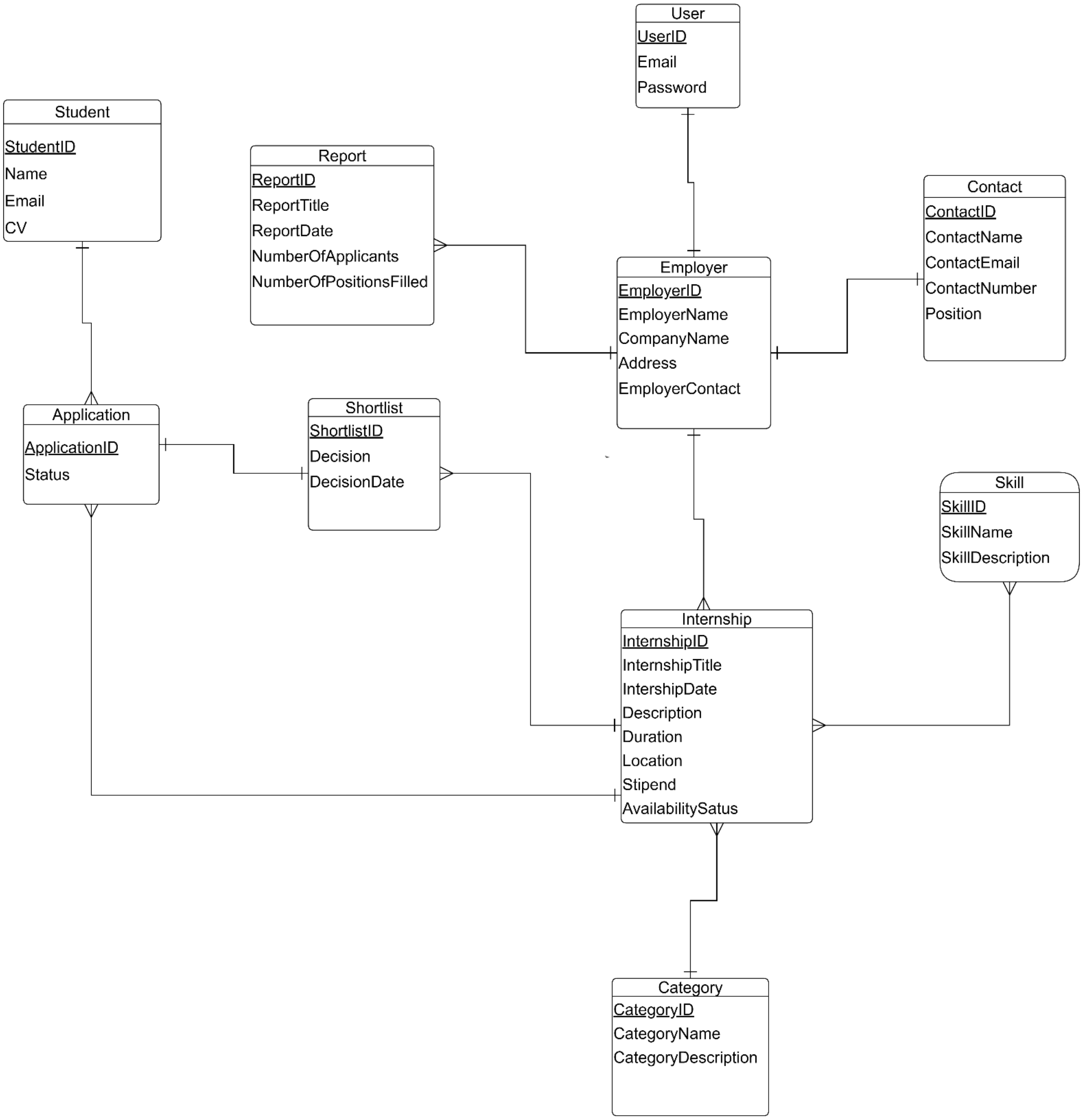
Entity Attributes List

1. User – UserID, Email, Password
2. Employer – EmployerID, EmployerName, CompanyName, Address, EmployerContact
3. Contact – ContactID, ContactName, ContactEmail, ContactNumber, Position
4. Skill – SkillID, SkillName, SkillDescription
5. Category – CategoryID, CategoryName, CategoryDescription
6. Internship – InternshipID, InternshipTitle, InternshipDate, Description, Duration, Location, Stipend, AvailabilityStatus
7. Student – StudentID, StudentName, Email, CV
8. Application - ApplicationID, ApplicationDate, Status
9. Shortlist – ShortlistID, Decision, DecisonDate
10. Report – ReportID, ReportTitle, ReportDate, NumberOfApplicants, NumberOfPositionFilled

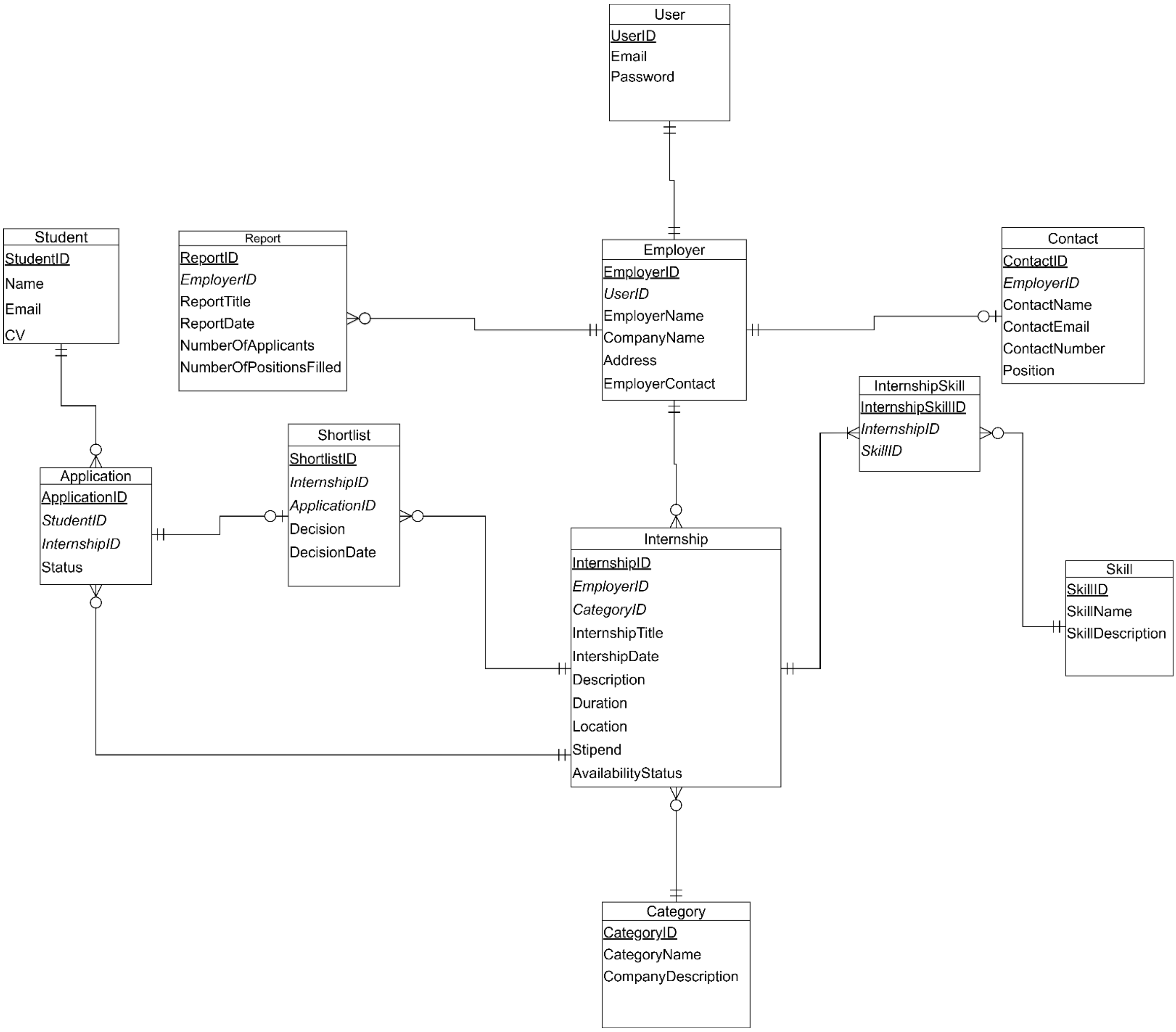
Entity Relationships

- User 1:1 Employer (User can be one Employer)
- Employer 1:1 Contact (Employer can create only one contact)
- Employer 1: M Internship (One employer can post many internships and each internship is posted by one employer.
- Internship M:M Skill (One internship can require multiple skills and one skill can be required by multiple internships.
- Internship M:1 Category (Many internships can belong to the same category and each internship belongs to one category.
- Internship 1:M Shortlist (One internship can have many shortlist records (for applicants) and each shortlist entry belongs to one internship.
- Employer 1:M Report (One employer can generate many reports and each report belongs to only one employer)
- Student 1:M Application (One Student may apply many applications)
- Shortlist 1: 1 Application (One application can have one shortlist entry)
- Internship 1:M Application (One internship can have many applications)

Logical ERD


User
UserID
Email
Password
Student
StudentID
Name
Email
CV
Report
ReportID
ReportTitle
ReportDate
NumberOfApplicants
NumberOfPositionsFilled
Contact
ContactID
ContactName
ContactEmail
ContactNumber
Position
Employer
EmployerID
EmployerName
CompanyName
Address
EmployerContact
Application
ApplicationID
Status
Shortlist
ShortlistID
Decision
DecisionDate
Skill
SkillID
SkillName
SkillDescription
Internship
InternshipID
InternshipTitle
IntershipDate
Description
Duration
Location
Stipend
AvailabilitySatus
Category
CategoryID
CategoryName
CategoryDescription

Physical ERD



Assumptions and Notes

User

- Can exist independently in the system (can signup without being an employer yet).

Employer

- Cannot exist without user account.

Contact

- Each contact must be linked to the employer.
- Cannot exist without the employer.

Internship

- Each Internship must belong to exactly one employer.
- Each Internship must belong to exactly one category.
- Cannot exist without Employer.

Skill

- Skills exist in the system independently.

InternshipSkill (Junction Table)

- Cannot exist without both a valid Internship and a valid Skill.
- Each internship must have at least one required skill, so for each Internship, at least one row exists.

Category

- Categories can exist without internships assigned yet.
- Once an internship is created, it must be assigned a category.

Shortlist

- Each Shortlist row must reference a valid Internship and Application.
- Shortlist entries are optional in the system until employer review applicants.

Report

- Reports cannot exist without an Employer.
- Employers may have zero or many reports.

Relational Schema

1. User – (UserID, Email, Password)
2. Employer – (EmployerID, *UserID*, EmployerName, CompanyName, Address, EmployerContact)
3. Contact – (ContactID, *EmployerID*, ContactName, ContactEmail, ContactNumber, Position)
4. Skill – (SkillID, SkillName, SkillDescription)
5. Category – (CategoryID, CategoryName, CategoryDescription)
6. Internship – (InternshipID, *EmployerID*, *CategoryID*, InternshipTitle, InternshipDate, Description, Duration, Location, Stipend, AvailabilityStatus)
7. InternshipSkill – (*InternshipID, SkillID*)
8. Student – (StudentID, StudentName, Email, CV)
9. Application – (ApplicationID, *StudentID*, InternshipID, ApplicationDate, Status)
10. Shortlist – (ShortlistID, *InternshipID*, *ApplicationID*, Decision, DecisonDate)
11. Report – (ReportID, *EmployerID*, ReportTitle, ReportDate, NumberOfApplicants, NumberOfPositionFilled)

Data Dictionary

a. User

Purpose: To store the login credentials of the Employer

Column Name	Data Type & Length	Null	Constraints	Other
UserID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
Email	VARCHAR (100)	NOT NULL	UNIQUE, CHECK (Email LIKE '%@%. %')	
Password	VARCHAR (225)	NOT NULL		

b. Employer

Purpose: To store Employer Details

Column Name	Data Type & Length	Null	Constraints	Other
EmployerID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
UserID	INT	NOT NULL	FK (User. UserID)	
EmployerName	VARCHAR (100)	NOT NULL		
CompanyName	VARCHAR (255)	NOT NULL		
Address	VARCHAR (255)	NOT NULL		
EmployerContact	VARCHAR (15)	NOT NULL		

c. Contact

Purpose: To store the details of the person who is handling Internships

Column Name	Data Type & Length	Null	Constraints	Other
ContactID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
EmployerID	INT	NOT NULL	FK (Employer. EmployerID)	
ContactName	VARCHAR (100)	NOT NULL		
CompanyEmail	VARCHAR (255)	NOT NULL	CHECK (Email LIKE '%@%. %')	
ContactNumber	VARCHAR (15)	NOT NULL		
Position	VARCHAR (100)	NOT NULL		

d. Skill

Purpose: To store IT Skills

Column Name	Data Type & Length	Null	Constraints	Other
SkillID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
SkillName	VARCHAR (100)	NOT NULL		
SkillDescription	VARCHAR (255)	NOT NULL		

e. Category

Purpose: To store Category of the Internship

Column Name	Data Type & Length	Null	Constraints	Other
CategoryID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
CategoryName	VARCHAR (100)	NOT NULL		
CategoryDescription	VARCHAR (255)	NOT NULL		

f. Internship

Purpose: To store Internship Details

Column Name	Data Type & Length	Null	Constraints	Other
InternshipID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
EmployerID	INT	NOT NULL	FK (Employer. EmployerID)	
CategoryID	INT	NOT NULL	FK (Category. CategoryID)	
InternshipTitle	VARCHAR (100)	NOT NULL		
InternshipDate	DATE	NOT NULL	DEFAULT = GETDATE ()	
Description	VARCHAR (255)	NOT NULL		
Duration	VARCHAR (100)	NOT NULL		
Location	VARCHAR (50)	NOT NULL	CHECK (Location IN ('On-Site', 'Remote', 'Hybrid'))	
Stipend	VARCHAR (50)	NOT NULL	CHECK (Stipend IN ('Paid', 'Unpaid'))	
Availability Status	VARCHAR (50)	NOT NULL	CHECK (AvailabilityStatus IN ('Open', 'Closed'))	

g. InternshipSkill (junction table)

Purpose: To have a junction table

Column Name	Data Type & Length	Null	Constraints	Other
InternshipID	INT	NOT NULL	PK (composite), FK → (Internship. InternshipID)	
SkillID	INT	NOT NULL	PK (composite), FK → (Skill. SkillID)	

h. Student

Purpose: To Use the details of the student

Column Name	Data Type & Length	Null	Constraints	Other
StudentID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
StudentName	VARCHAR (100)	NOT NULL		
Email	INT	NOT NULL	UNIQUE	
CV	VARCHAR (50)	NOT NULL		

i. Application

Purpose: To get the applications to shortlist applicants

Column Name	Data Type & Length	Null	Constraints	Other
ApplicationID	INT	NOT NULL	PK. UNIQUE	IDENTITY (Auto Increment)
StudentID	INT	NOT NULL	FK (Student. StudentID)	
InternshipID	INT	NOT NULL	FK (Internship. InternshipID)	
ApplicationDate	DATE	NOT NULL		DEFAULT = GETDATE ()
Status	VARCHAR (50)	NOT NULL		

j. Shortlist

Purpose: To store decision of the Applicant

Column Name	Data Type & Length	Null	Constraints	Other
ShorlistID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
InternshipID	INT	NOT NULL	FK (Internship. InternshipID)	
ApplicationID	INT	NOT NULL	FK (Application. ApplicationID)	
Decision	VARCHAR (50)	NOT NULL	CHECK (Decision IN ('Shortlisted', 'Rejected'))	
DecisionDate	DATE	NOT NULL	DEFAULT = GETDATE ()	

k. Report

Purpose to generate reports for internships at a given period

Column Name	Data Type & Length	Null	Constraints	Other
ReportID	INT	NOT NULL	PK, UNIQUE	IDENTITY (Auto Increment)
EmployerID	INT	NOT NULL	FK (Employer. EmployerID)	
ReportTitle	VARCHAR (100)	NOT NULL		
ReportDate	DATE	NOT NULL	DEFAULT = _GETDATE ()	
NumberOfApplicants	INT	NOT NULL	DEFAULT 0	
NumberOfPostitionsFilled	INT	NOT NULL	DEFAULT 0	

Normalization

1. User (UserID, Email, Password)

    i. Primary Key – UserID

    ii. FDs – UserID → Email, Password

    iii. 1NF: All attributes are atomic, no repeating groups

    iv. 2NF: Single-Column key, no partial dependencies

    v. 3NF: All the attributes depend on the primary key.

2. Employer – (EmployerID, *UserID*, EmployerName, CompanyName, Address, EmployerContact)

    i. Primary Key – EmployerID

    ii. FDs – EmployerID → UserID, EmployerName, CompanyName, Address, EmployerContact)

    iii. 1NF: All atomic, one row per employer

    iv. 2NF: Single-Column key, no partial dependencies

    v. 3NF: All non-keys depend entirely on EmployerID.

3. Contact – (ContactID, *EmployerID*, ContactName, ContactEmail, ContactNumber, Position)

    i. Primary Key – ContactID

    ii. FDs – ContactID → EmployerID, ContactName, ContactEmail, ContactNumber, Position)

    iii. 1NF: All atomic fields

    iv. 2NF: Single Primary Key, no partial dependencies

    v. 3NF: No non-key determines another non-key (e.g., CompanyEmail does not determine Position)

4. Skill – (<u>SkillID</u>, SkillName, SkillDescription)

    i. Primary Key – SkillID
    ii. FDs – SkillID → (SkillName, SkillDescription)
    iii. 1NF: Atomic Values
    iv. 2NF: Single Primary Key, no partial dependency
    v. 3NF: No transitive dependencies

5. Category - (<u>CategoryID</u>, CategoryName, CategoryDescription)

    i. Primary Key – CategoryID
    ii. FDs – CategoryID → CategoryName, CategoryDescription
    iii. 1NF: Atomic Values
    iv. 2NF: Single Primary Key, no partial dependency
    v. 3NF: No transitive dependencies

6. Internship - (<u>InternshipID</u>, *EmployerID*, *CategoryID*, InternshipTitle, InternshipDate, Description, Duration, Location, Stipend, AvailabilityStatus)

    i. Primary Key – InternshipID
    ii. FDs – InternshipID → EmployerID, CategoryID, InternshipTitle, InternshipDate, Description, Duration, Location, Stipend, AvailabilityStatus)
    iii. 1NF: All columns are atomic (e.g., Duration is a single value like “3 months”)
    iv. 2NF: Single Primary Key, no partial dependencies
    v. 3NF: No non-key determines another non-key

7. InternshipSkill – (*InternshipID*, *SkillID*, PK(InternshipID, SkillID))

   i. Primary Key – (InternshipID, SkillID)
   ii. FDs – (InternshipID, SkillID) → none
   iii. 1NF: Each row records one Internship–Skill pairing: atomic
   iv. 2NF: With composite key, check partial dependencies, there are no on keys. No partial dependencies
   v. 3NF: No transitive dependencies

8. Student - (StudentID, StudentName, Email, CV)

   i. Primary Key – (StudentID)
   ii. FDs- StudentID → (StudentName, Email, CV)
   iii. 1NF: Atomic Values
   iv. 2NF: Single Primary Key, no partial dependency
   v. 3NF: No transitive dependencies

9. Application - (ApplicationID, *StudentID*, InternshipID, ApplicationDate, Status)

   i. Primary Key – (ApplicationID)
   ii. FDs- ApplicationID → (StudentID, InternshipID, ApplicationDate, Status)
   iii. 1NF: Atomic Values
   iv. 2NF: Single Primary Key, no partial dependency (Foreign keys depends on the ApplicationID)
   v. 3NF: No transitive dependencies

10. Shortlist - (ShortlistID, *InternshipID*, *ApplicationID*, Decision, DecisonDate)
    i. Primary Key – ShortlistID
    ii. FDs – ShortlistID → InternshipID, ApplicationID, Decision, DecisonDate
    iii. 1NF: Atomic Values
    iv. 2NF: Single Primary Key, no partial dependency
    v. 3NF: Decision and DecisionDate depend on the key only

11. Report - (ReportID, *EmployerID*, ReportTitle, ReportDate, NumberOfApplicants, NumberOfPositionFilled)
    i. Primary Key – ReportID
    ii. FDs – ReportID → EmployerID, ReportTitle, ReportDate, NumberOfApplicants, NumberOfPositionFilled
    iii. 1NF: Atomic Values. Each row summarizes a period
    iv. 2NF: Single Primary Key, no partial dependency
    v. 3NF: No non-key determines another non-key under the stated design.

Research on NoSQL

Massive amounts of unstructured or semi-structured data can be handled, stored, and retrieved using relational database systems called NoSQL (Not Only SQL) databases. NoSQL databases are more adaptable and without structures than relational databases built on organized query language (SQL) and fixed schemas.

NoSQL was born in the late 2000s as a reaction to the increasing need of dealing with big data, real-time web applications, and scalable distributed systems. Google, Amazon, and Facebook were some of the organizations finding it difficult to scale relational databases to deal with billions of transactions and massive amounts of user-generated content. Therefore, NoSQL emerged as an approach that is more concerned with scalability, performance, and flexibility than with rigid relational integrity.

Applications of NoSQL

NoSQL databases are employed heavily in all the places where flexibility and scalability are given the highest priority:

- **Social Media Websites** – User data, feeds, and messaging data (e.g., Facebook, Twitter).
- **E-Commerce** – Product database, customer data, and recommendations (e.g., Amazon).
- **Content Management Systems** – Blogs, videos, and multimedia storage.
- **Big Data and Analytics** – Real-time processing of data in IoT and machine learning.
- **Financial Services** – Fraud identification and high-frequency trading where consumption of data should be carried out in real-time.

Suitability of NoSQL

NoSQL databases are more suitable for:

- applications that handle massive amounts of unstructured or quickly changing data.
- Systems must be able to extend horizontally, meaning they must add servers rather than upgrade existing ones.
- programs that operate in real time and put speed ahead of solid consistency.
- projects where tight schemas cannot limit the data models, which are often updated.

Limitations of NoSQL

- Weaker Consistency: Rather of complying strictly to ACID compliance, NoSQL databases use eventual consistency.
- Limited Query Support: NoSQL systems lack the robustness of relational databases when dealing with to complex joins and transactions
- Less Mature Ecosystem: Compared to SQL databases, NoSQL tools and standards are more recently developed.
- Learning Curve: Developers who are used to SQL may find it challenging to adapt towards schema-less designs.

Examples for NoSQL Databases: MongoDB, CouchDB, Redis, DynamoDB etc.…

MongoDB vs MYSQL

Aspect	MongoDB	MySQL
Performance	Designed for high-speed read and write of vast volumes of unstructured or semi-structured data. It works well in the scenario of simple queries and sharded data.	Very efficient for structured queries, joined queries, and transactions. It becomes slow with enormous amounts of unstructured data or frequent schema updates.
Scalability	Executed using horizontal scaling (sharding and replication across servers). Suitable for high-traffic cloud-based applications.	Exposed primarily to vertical scaling (upgrading hardware). Horizontal scaling is not offered but more complex and less fluid than MongoDB.
Flexibility	Schema-less design allows dynamic addition of fields without changing the database schema. Ideal for rapid-changing applications (e.g., startups, agile projects).	Schema-dependent; any change entails altering the schema, which is time-consuming. More restrictive but ensures good data integrity.

Benefits and Trade-offs

Benefits of MongoDB

- **High Scalability:** Most suitable for highly visited websites because it is horizontally scalable with sharding.
- **Schema Flexibility:** Temporarily fast-moving and quickly changing projects are supported using data structures that can change without schema migrations.
- **Performance on Big Data:** Structured to read and write vast amounts of unstructured or semi-structured data at a high speed.
- **Cloud-Native:** Suitable Cloud deployments and distributed systems

Trade-offs of MongoDB

- **Weaker ACID Compliance:** sets scalability and performance ahead of strong consistency (eventual consistency is common).
- **Restricted Complex Queries:** In compared to SQL databases, it does not provide solid assistance for joins or complex relational queries.

- **Less Mature Ecosystem:** The tooling and ecosystem are still less extensive than traditional relational databases, despite their growth.

Benefits of MySQL

Strong ACID Properties: Ensures data integrity and consistency, as required for transactional workloads.
Efficient Structured Queries: Works well for relational queries and complex joins.
Mature Ecosystem: Active community support, docs, and tooling.
Stability and Reliability: Stable technology running in hundreds of production stacks.

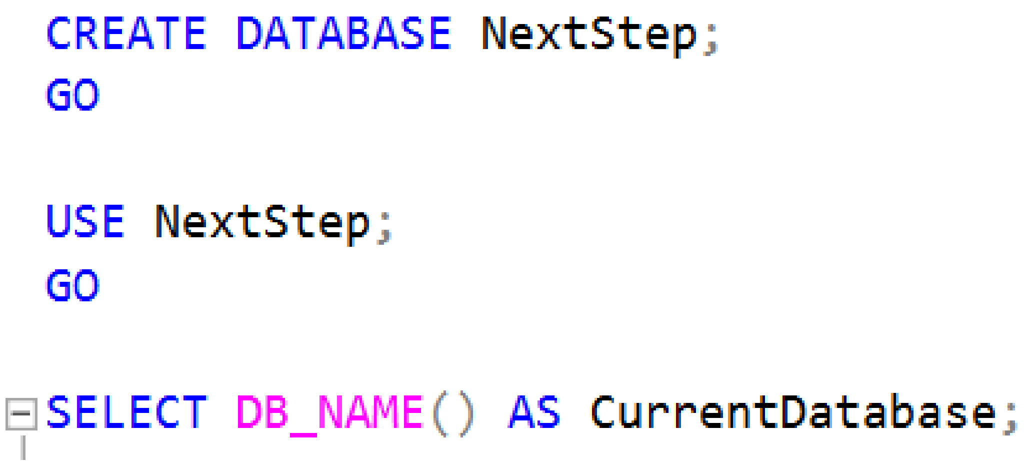
Trade- offs of MySQL

Strict Schema: Schema modifications will require migrations, which are invasive and time-consuming.
Limited Scalability: Scaling is primarily vertical where Horizontal scaling is more complex compared to NoSQL solutions.
Performance Bottlenecks with Big Data: Can become slow when handling large, unstructured, or highly changing data sets.

Conclusion

MySQL is more often used for web applications with structured data for strong transactional needs while MongoDB can be used for scalable, data-intensive or rapidly evolving applications for a better performance. The choice is depending on the application requirements.

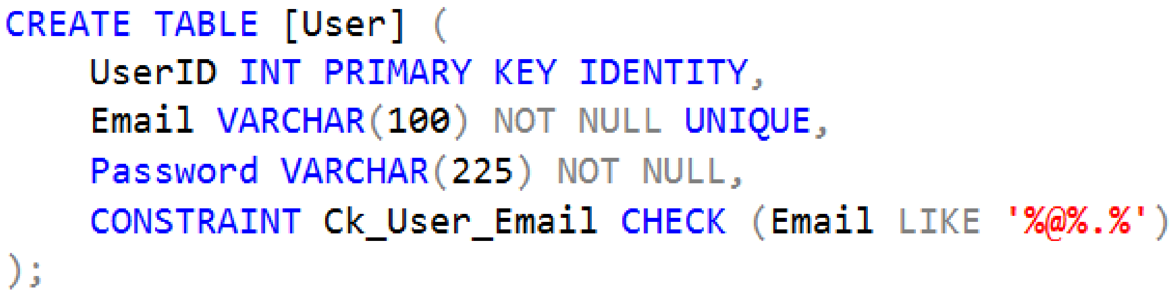
SQL Database Creation



```
CREATE DATABASE NextStep;
GO

USE NextStep;
GO

SELECT DB_NAME() AS CurrentDatabase;
```

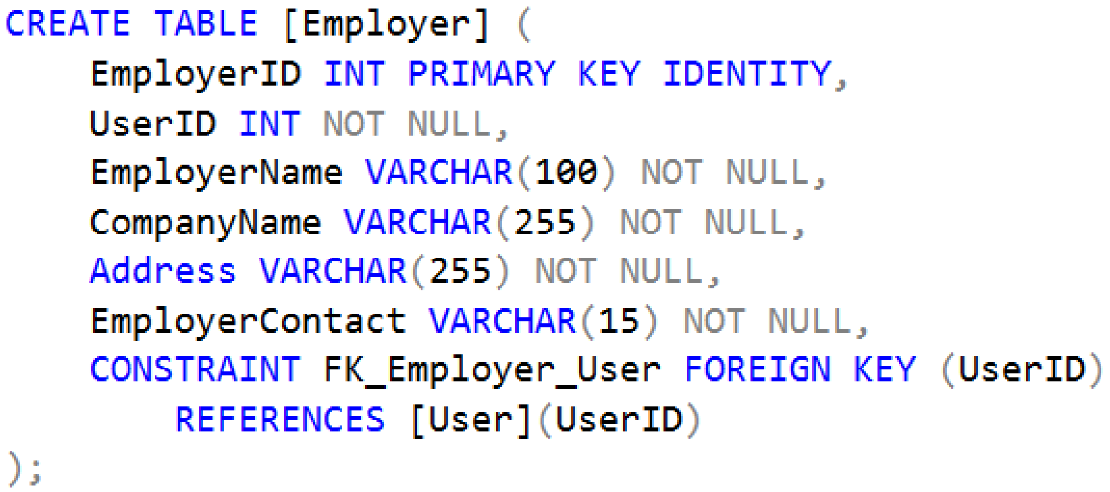
Table Creation

User Table



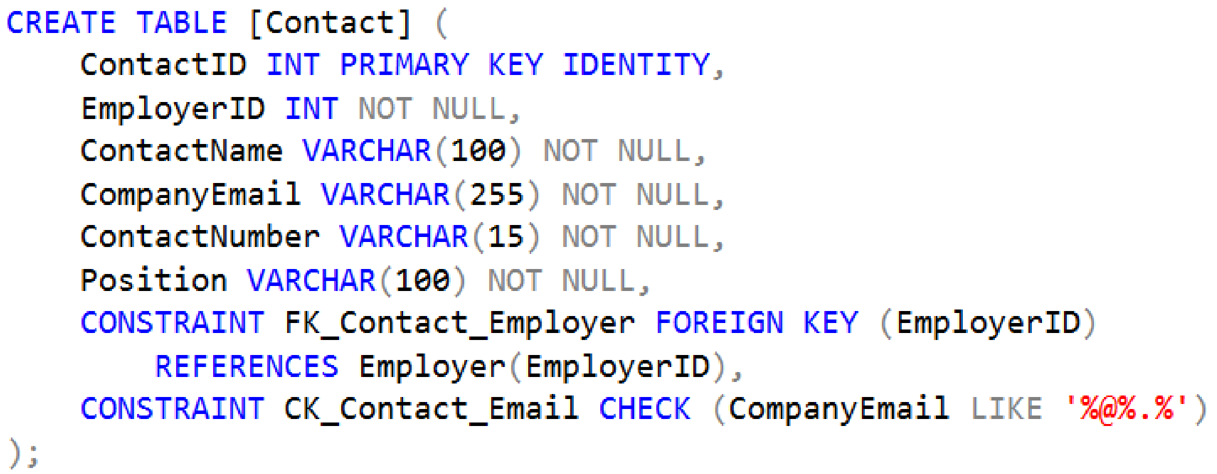
```
CREATE TABLE [User] (
    UserID INT PRIMARY KEY IDENTITY,
    Email VARCHAR(100) NOT NULL UNIQUE,
    Password VARCHAR(225) NOT NULL,
    CONSTRAINT Ck_User_Email CHECK (Email LIKE '%@%.%')
);
```

Employer Table



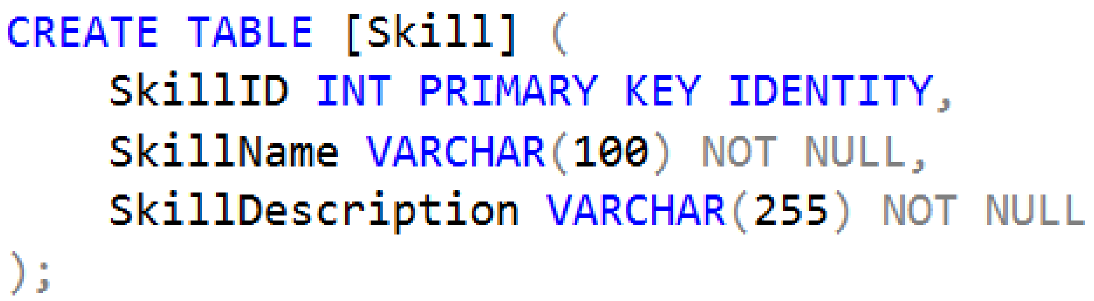
```
CREATE TABLE [Employer] (
    EmployerID INT PRIMARY KEY IDENTITY,
    UserID INT NOT NULL,
    EmployerName VARCHAR(100) NOT NULL,
    CompanyName VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    EmployerContact VARCHAR(15) NOT NULL,
    CONSTRAINT FK_Employer_User FOREIGN KEY (UserID)
        REFERENCES [User](UserID)
);
```

Contact Table



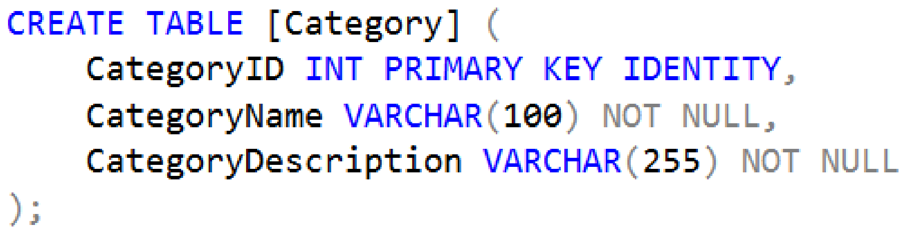
```
CREATE TABLE [Contact] (
    ContactID INT PRIMARY KEY IDENTITY,
    EmployerID INT NOT NULL,
    ContactName VARCHAR(100) NOT NULL,
    CompanyEmail VARCHAR(255) NOT NULL,
    ContactNumber VARCHAR(15) NOT NULL,
    Position VARCHAR(100) NOT NULL,
    CONSTRAINT FK_Contact_Employer FOREIGN KEY (EmployerID)
        REFERENCES Employer(EmployerID),
    CONSTRAINT CK_Contact_Email CHECK (CompanyEmail LIKE '%@%.%')
);
```

Skill Table



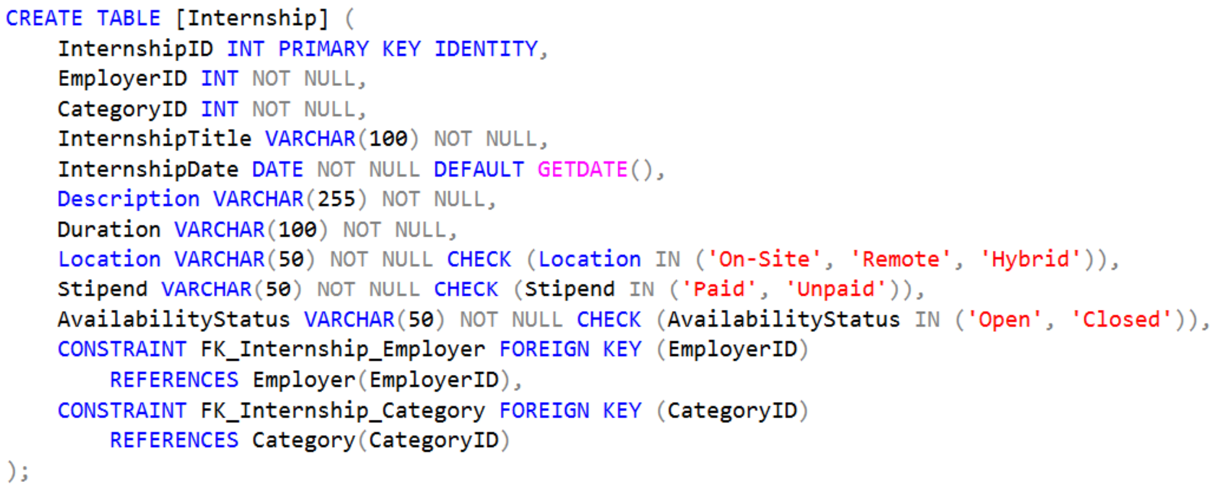
```
CREATE TABLE [Skill] (
    SkillID INT PRIMARY KEY IDENTITY,
    SkillName VARCHAR(100) NOT NULL,
    SkillDescription VARCHAR(255) NOT NULL
);
```

Category Table



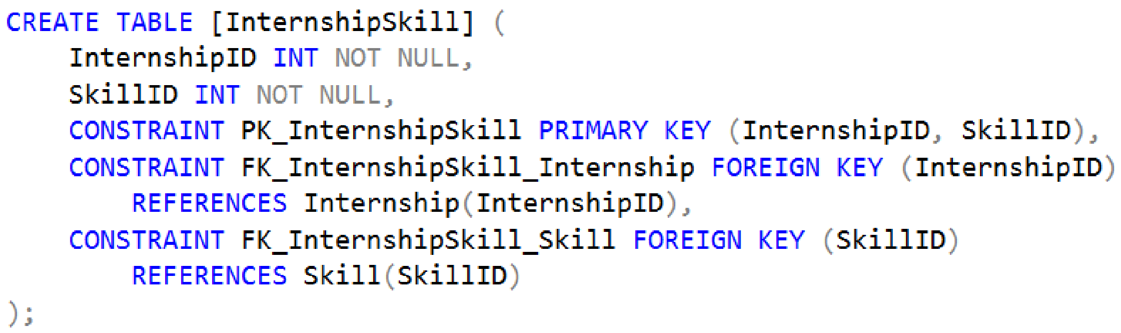
```
CREATE TABLE [Category] (
    CategoryID INT PRIMARY KEY IDENTITY,
    CategoryName VARCHAR(100) NOT NULL,
    CategoryDescription VARCHAR(255) NOT NULL
);
```

Internship Table



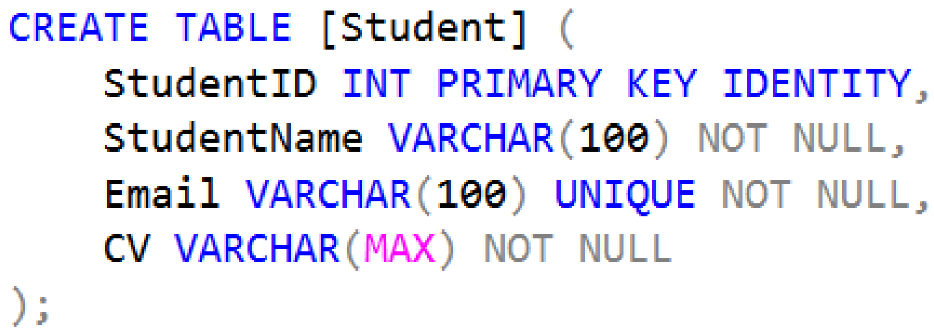
```sql
CREATE TABLE [Internship] (
    InternshipID INT PRIMARY KEY IDENTITY,
    EmployerID INT NOT NULL,
    CategoryID INT NOT NULL,
    InternshipTitle VARCHAR(100) NOT NULL,
    InternshipDate DATE NOT NULL DEFAULT GETDATE(),
    Description VARCHAR(255) NOT NULL,
    Duration VARCHAR(100) NOT NULL,
    Location VARCHAR(50) NOT NULL CHECK (Location IN ('On-Site', 'Remote', 'Hybrid')),
    Stipend VARCHAR(50) NOT NULL CHECK (Stipend IN ('Paid', 'Unpaid')),
    AvailabilityStatus VARCHAR(50) NOT NULL CHECK (AvailabilityStatus IN ('Open', 'Closed')),
    CONSTRAINT FK_Internship_Employer FOREIGN KEY (EmployerID)
        REFERENCES Employer(EmployerID),
    CONSTRAINT FK_Internship_Category FOREIGN KEY (CategoryID)
        REFERENCES Category(CategoryID)
);
```

InternshipSkill (Junction Table)



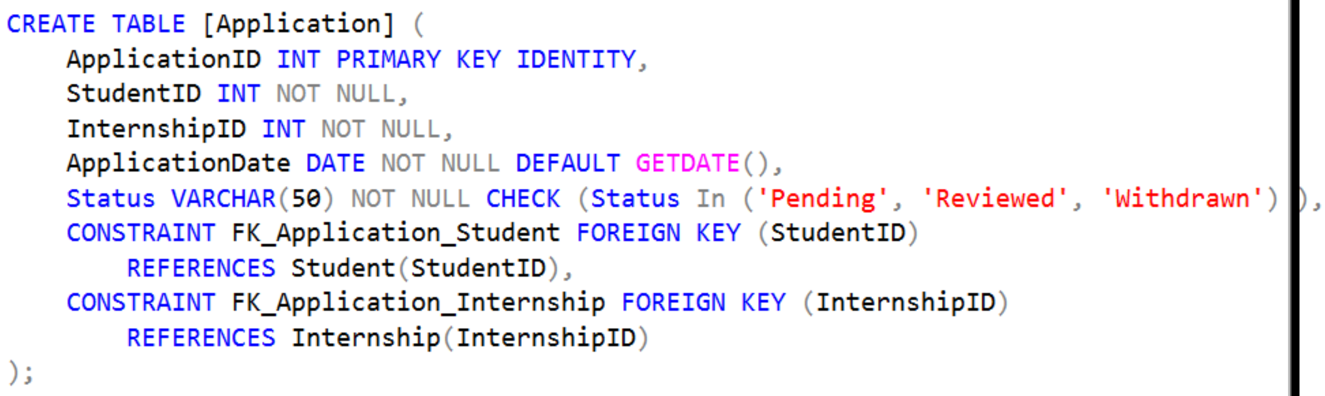
```sql
CREATE TABLE [InternshipSkill] (
    InternshipID INT NOT NULL,
    SkillID INT NOT NULL,
    CONSTRAINT PK_InternshipSkill PRIMARY KEY (InternshipID, SkillID),
    CONSTRAINT FK_InternshipSkill_Internship FOREIGN KEY (InternshipID)
        REFERENCES Internship(InternshipID),
    CONSTRAINT FK_InternshipSkill_Skill FOREIGN KEY (SkillID)
        REFERENCES Skill(SkillID)
);
```

Student Table



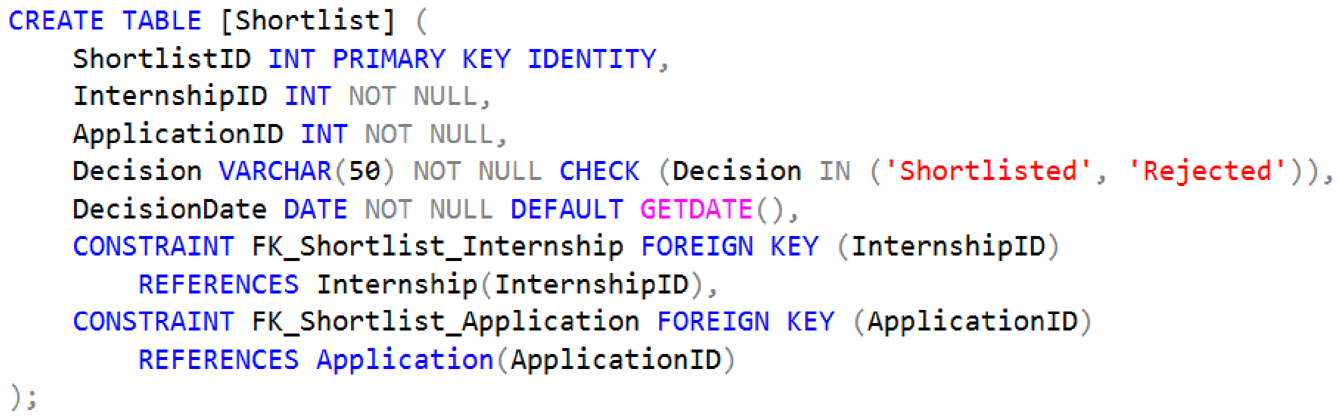
```
CREATE TABLE [Student] (
    StudentID INT PRIMARY KEY IDENTITY,
    StudentName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    CV VARCHAR(MAX) NOT NULL
);
```

Application Table



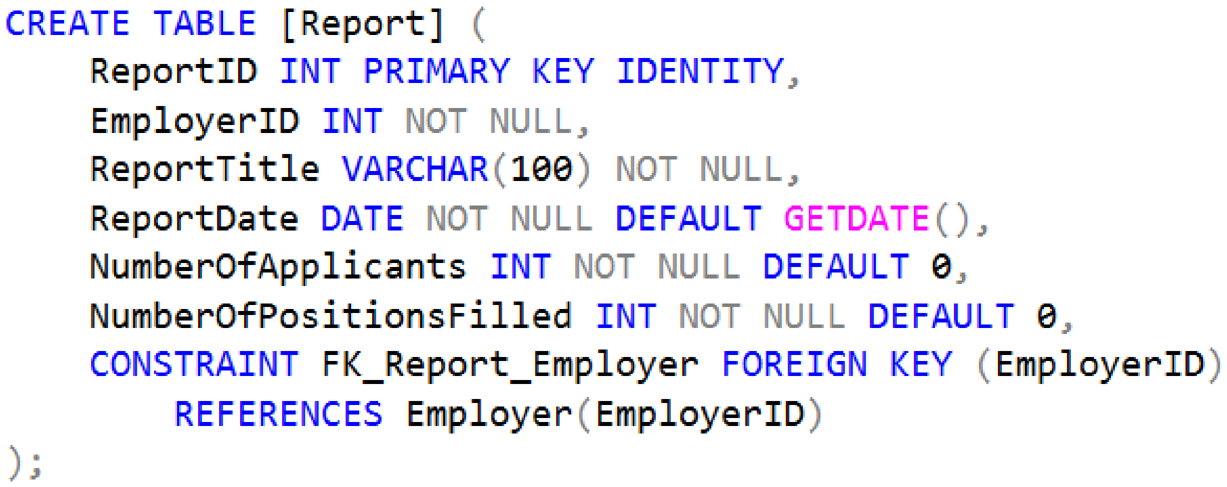
```
CREATE TABLE [Application] (
    ApplicationID INT PRIMARY KEY IDENTITY,
    StudentID INT NOT NULL,
    InternshipID INT NOT NULL,
    ApplicationDate DATE NOT NULL DEFAULT GETDATE(),
    Status VARCHAR(50) NOT NULL CHECK (Status In ('Pending', 'Reviewed', 'Withdrawn')),
    CONSTRAINT FK_Application_Student FOREIGN KEY (StudentID)
        REFERENCES Student(StudentID),
    CONSTRAINT FK_Application_Internship FOREIGN KEY (InternshipID)
        REFERENCES Internship(InternshipID)
);
```

Shortlist Table



```
CREATE TABLE [Shortlist] (
    ShortlistID INT PRIMARY KEY IDENTITY,
    InternshipID INT NOT NULL,
    ApplicationID INT NOT NULL,
    Decision VARCHAR(50) NOT NULL CHECK (Decision IN ('Shortlisted', 'Rejected')),
    DecisionDate DATE NOT NULL DEFAULT GETDATE(),
    CONSTRAINT FK_Shortlist_Internship FOREIGN KEY (InternshipID)
        REFERENCES Internship(InternshipID),
    CONSTRAINT FK_Shortlist_Application FOREIGN KEY (ApplicationID)
        REFERENCES Application(ApplicationID)
);
```

Report Table

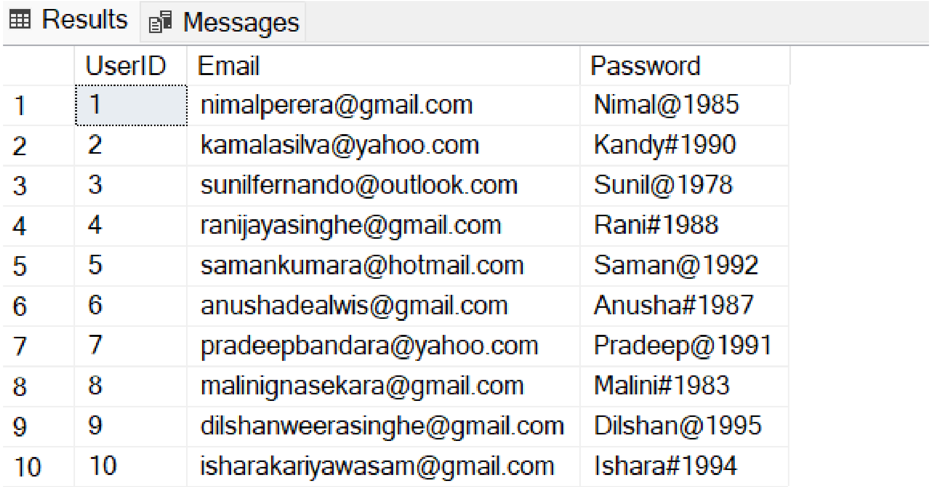


```
CREATE TABLE [Report] (
    ReportID INT PRIMARY KEY IDENTITY,
    EmployerID INT NOT NULL,
    ReportTitle VARCHAR(100) NOT NULL,
    ReportDate DATE NOT NULL DEFAULT GETDATE(),
    NumberOfApplicants INT NOT NULL DEFAULT 0,
    NumberOfPositionsFilled INT NOT NULL DEFAULT 0,
    CONSTRAINT FK_Report_Employer FOREIGN KEY (EmployerID)
        REFERENCES Employer(EmployerID)
);
```

Data Insertion

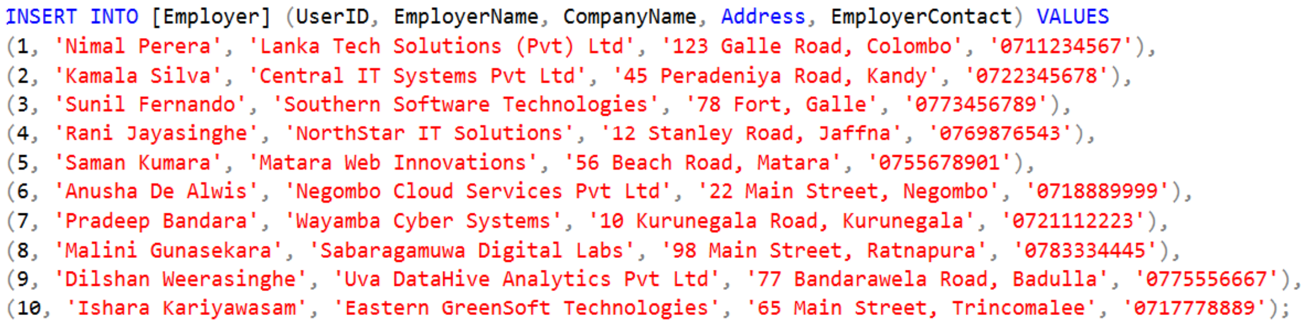
User Table

```sql
INSERT INTO [User] (Email, Password) VALUES
('nimalperera@gmail.com', 'Nimal@1985'),
('kamalasilva@yahoo.com', 'Kandy#1990'),
('sunilfernando@outlook.com', 'Sunil@1978'),
('ranijayasinghe@gmail.com', 'Rani#1988'),
('samankumara@hotmail.com', 'Saman@1992'),
('anushadealwis@gmail.com', 'Anusha#1987'),
('pradeepbandara@yahoo.com', 'Pradeep@1991'),
('malinignasekara@gmail.com', 'Malini#1983'),
('dilshanweerasinghe@gmail.com', 'Dilshan@1995'),
('isharakariyawasam@gmail.com', 'Ishara#1994');
```

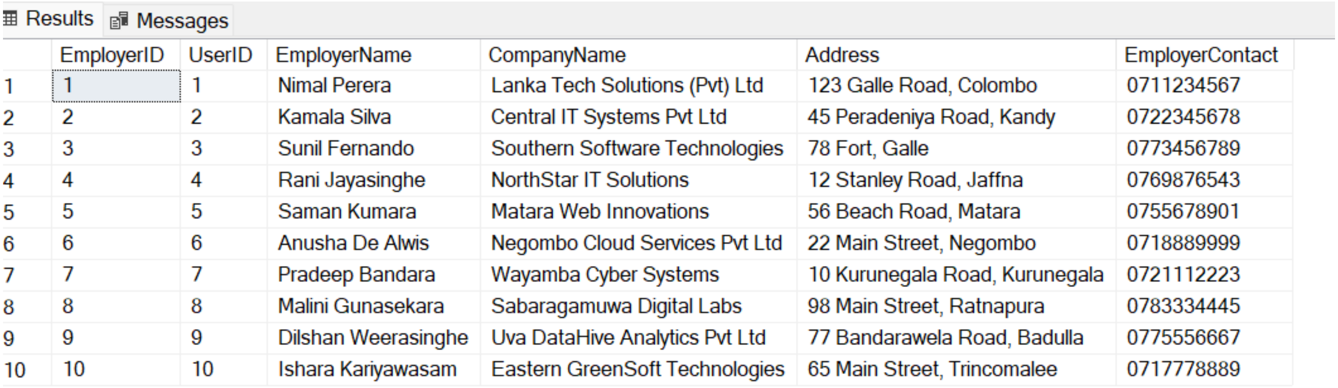


	UserID	Email	Password
1	1	nimalperera@gmail.com	Nimal@1985
2	2	kamalasilva@yahoo.com	Kandy#1990
3	3	sunilfernando@outlook.com	Sunil@1978
4	4	ranijayasinghe@gmail.com	Rani#1988
5	5	samankumara@hotmail.com	Saman@1992
6	6	anushadealwis@gmail.com	Anusha#1987
7	7	pradeepbandara@yahoo.com	Pradeep@1991
8	8	malinignasekara@gmail.com	Malini#1983
9	9	dilshanweerasinghe@gmail.com	Dilshan@1995
10	10	isharakariyawasam@gmail.com	Ishara#1994

Employer Table

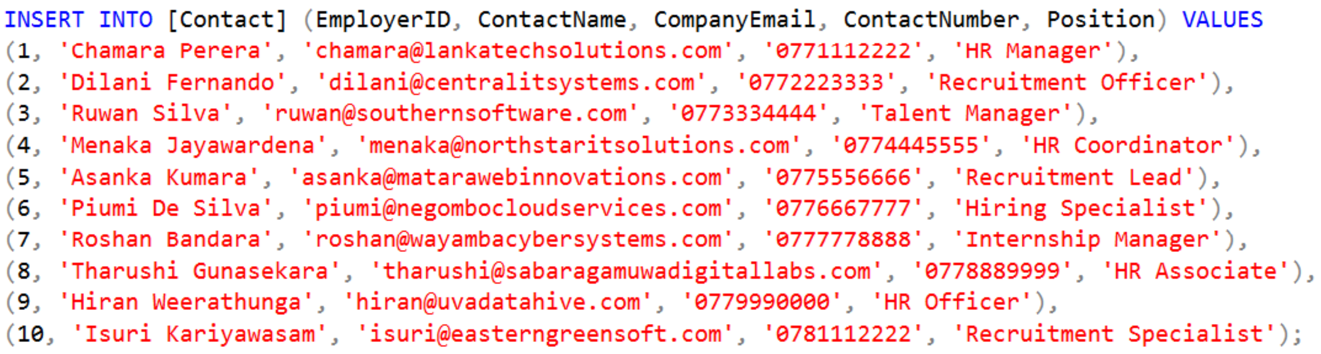


```sql
INSERT INTO [Employer] (UserID, EmployerName, CompanyName, Address, EmployerContact) VALUES
(1, 'Nimal Perera', 'Lanka Tech Solutions (Pvt) Ltd', '123 Galle Road, Colombo', '0711234567'),
(2, 'Kamala Silva', 'Central IT Systems Pvt Ltd', '45 Peradeniya Road, Kandy', '0722345678'),
(3, 'Sunil Fernando', 'Southern Software Technologies', '78 Fort, Galle', '0773456789'),
(4, 'Rani Jayasinghe', 'NorthStar IT Solutions', '12 Stanley Road, Jaffna', '0769876543'),
(5, 'Saman Kumara', 'Matara Web Innovations', '56 Beach Road, Matara', '0755678901'),
(6, 'Anusha De Alwis', 'Negombo Cloud Services Pvt Ltd', '22 Main Street, Negombo', '0718889999'),
(7, 'Pradeep Bandara', 'Wayamba Cyber Systems', '10 Kurunegala Road, Kurunegala', '0721112223'),
(8, 'Malini Gunasekara', 'Sabaragamuwa Digital Labs', '98 Main Street, Ratnapura', '0783334445'),
(9, 'Dilshan Weerasinghe', 'Uva DataHive Analytics Pvt Ltd', '77 Bandarawela Road, Badulla', '0775556667'),
(10, 'Ishara Kariyawasam', 'Eastern GreenSoft Technologies', '65 Main Street, Trincomalee', '0717778889');
```

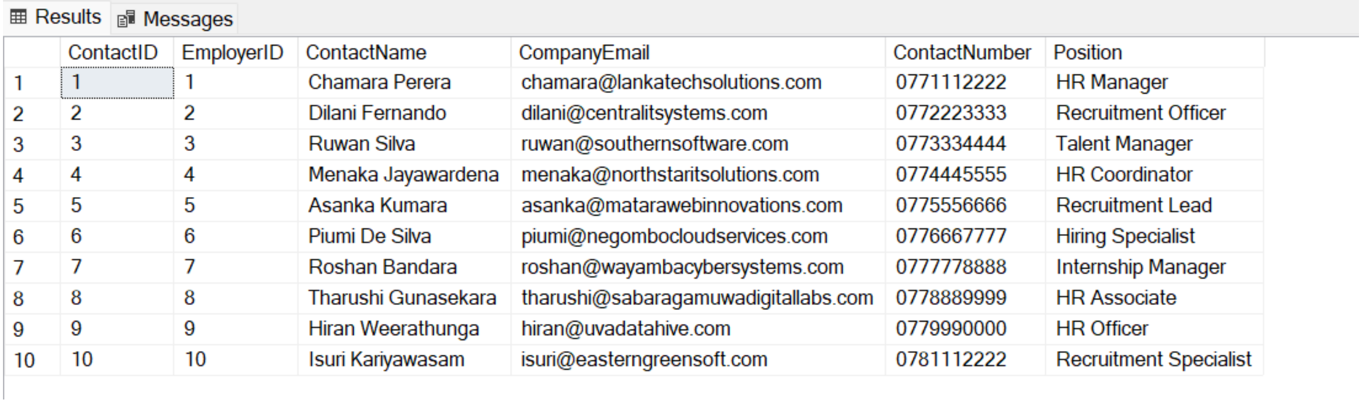


	EmployerID	UserID	EmployerName	CompanyName	Address	EmployerContact
1	1	1	Nimal Perera	Lanka Tech Solutions (Pvt) Ltd	123 Galle Road, Colombo	0711234567
2	2	2	Kamala Silva	Central IT Systems Pvt Ltd	45 Peradeniya Road, Kandy	0722345678
3	3	3	Sunil Fernando	Southern Software Technologies	78 Fort, Galle	0773456789
4	4	4	Rani Jayasinghe	NorthStar IT Solutions	12 Stanley Road, Jaffna	0769876543
5	5	5	Saman Kumara	Matara Web Innovations	56 Beach Road, Matara	0755678901
6	6	6	Anusha De Alwis	Negombo Cloud Services Pvt Ltd	22 Main Street, Negombo	0718889999
7	7	7	Pradeep Bandara	Wayamba Cyber Systems	10 Kurunegala Road, Kurunegala	0721112223
8	8	8	Malini Gunasekara	Sabaragamuwa Digital Labs	98 Main Street, Ratnapura	0783334445
9	9	9	Dilshan Weerasinghe	Uva DataHive Analytics Pvt Ltd	77 Bandarawela Road, Badulla	0775556667
10	10	10	Ishara Kariyawasam	Eastern GreenSoft Technologies	65 Main Street, Trincomalee	0717778889

Contact Table

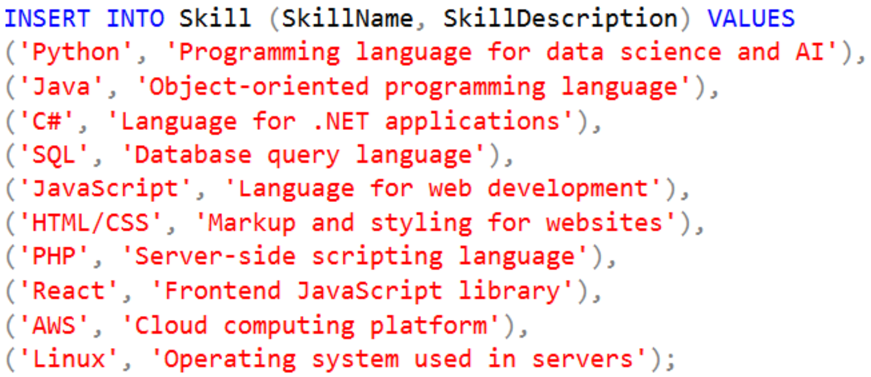


```sql
INSERT INTO [Contact] (EmployerID, ContactName, CompanyEmail, ContactNumber, Position) VALUES
(1, 'Chamara Perera', 'chamara@lankatechsolutions.com', '0771112222', 'HR Manager'),
(2, 'Dilani Fernando', 'dilani@centralitsystems.com', '0772223333', 'Recruitment Officer'),
(3, 'Ruwan Silva', 'ruwan@southernsoftware.com', '0773334444', 'Talent Manager'),
(4, 'Menaka Jayawardena', 'menaka@northstaritsolutions.com', '0774445555', 'HR Coordinator'),
(5, 'Asanka Kumara', 'asanka@matarawebinnovations.com', '0775556666', 'Recruitment Lead'),
(6, 'Piumi De Silva', 'piumi@negombocloudservices.com', '0776667777', 'Hiring Specialist'),
(7, 'Roshan Bandara', 'roshan@wayambacybersystems.com', '0777778888', 'Internship Manager'),
(8, 'Tharushi Gunasekara', 'tharushi@sabaragamuwadigitallabs.com', '0778889999', 'HR Associate'),
(9, 'Hiran Weerathunga', 'hiran@uvadatahive.com', '0779990000', 'HR Officer'),
(10, 'Isuri Kariyawasam', 'isuri@easterngreensoft.com', '0781112222', 'Recruitment Specialist');
```

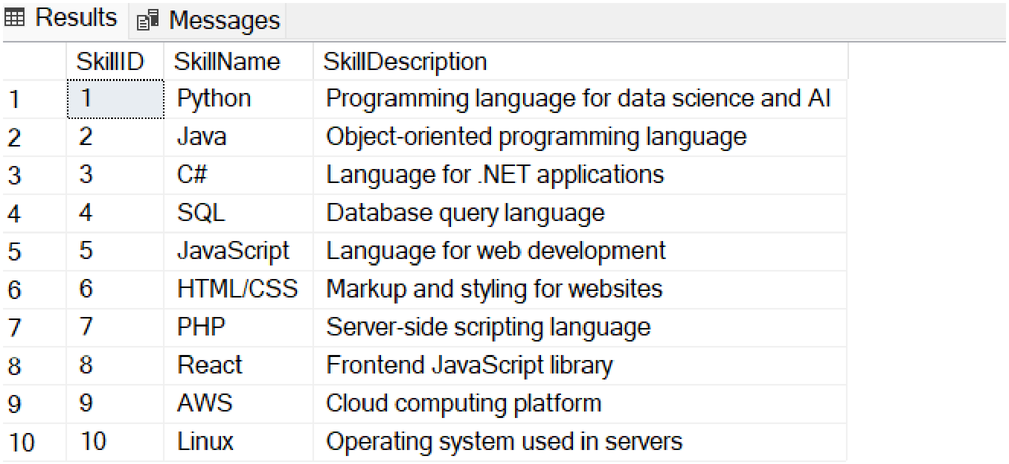


	ContactID	EmployerID	ContactName	CompanyEmail	ContactNumber	Position
1	1	1	Chamara Perera	chamara@lankatechsolutions.com	0771112222	HR Manager
2	2	2	Dilani Fernando	dilani@centralitsystems.com	0772223333	Recruitment Officer
3	3	3	Ruwan Silva	ruwan@southernsoftware.com	0773334444	Talent Manager
4	4	4	Menaka Jayawardena	menaka@northstaritsolutions.com	0774445555	HR Coordinator
5	5	5	Asanka Kumara	asanka@matarawebinnovations.com	0775556666	Recruitment Lead
6	6	6	Piumi De Silva	piumi@negombocloudservices.com	0776667777	Hiring Specialist
7	7	7	Roshan Bandara	roshan@wayambacybersystems.com	0777778888	Internship Manager
8	8	8	Tharushi Gunasekara	tharushi@sabaragamuwadigitallabs.com	0778889999	HR Associate
9	9	9	Hiran Weerathunga	hiran@uvadatahive.com	0779990000	HR Officer
10	10	10	Isuri Kariyawasam	isuri@easterngreensoft.com	0781112222	Recruitment Specialist

Skill Table



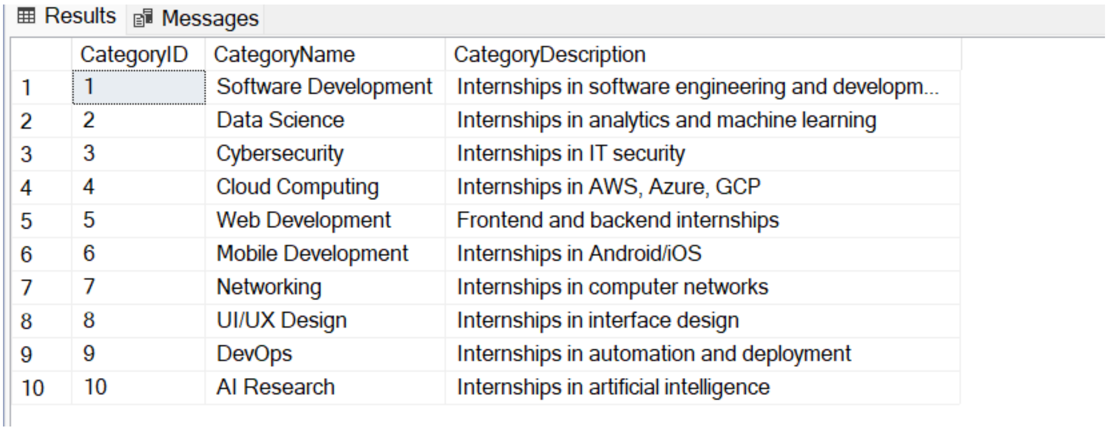
```sql
INSERT INTO Skill (SkillName, SkillDescription) VALUES
('Python', 'Programming language for data science and AI'),
('Java', 'Object-oriented programming language'),
('C#', 'Language for .NET applications'),
('SQL', 'Database query language'),
('JavaScript', 'Language for web development'),
('HTML/CSS', 'Markup and styling for websites'),
('PHP', 'Server-side scripting language'),
('React', 'Frontend JavaScript library'),
('AWS', 'Cloud computing platform'),
('Linux', 'Operating system used in servers');
```



	SkillID	SkillName	SkillDescription
1	1	Python	Programming language for data science and AI
2	2	Java	Object-oriented programming language
3	3	C#	Language for .NET applications
4	4	SQL	Database query language
5	5	JavaScript	Language for web development
6	6	HTML/CSS	Markup and styling for websites
7	7	PHP	Server-side scripting language
8	8	React	Frontend JavaScript library
9	9	AWS	Cloud computing platform
10	10	Linux	Operating system used in servers

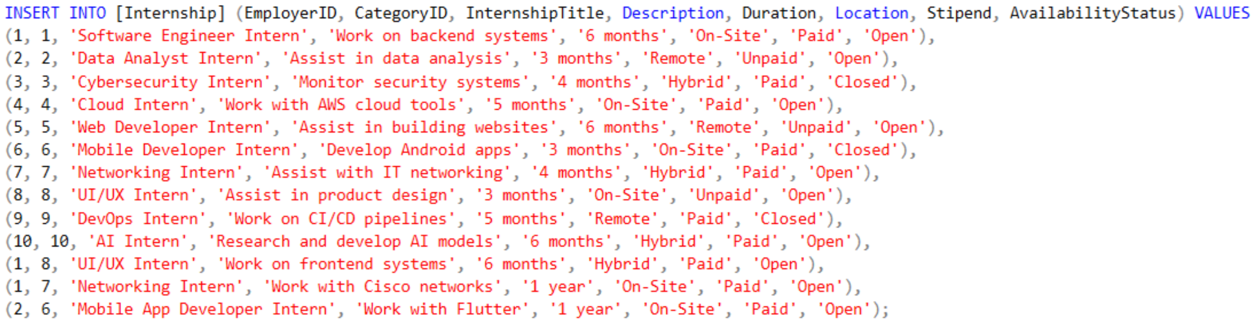
Category Table

```
INSERT INTO [Category] (CategoryName, CategoryDescription) VALUES
('Software Development', 'Internships in software engineering and development'),
('Data Science', 'Internships in analytics and machine learning'),
('Cybersecurity', 'Internships in IT security'),
('Cloud Computing', 'Internships in AWS, Azure, GCP'),
('Web Development', 'Frontend and backend internships'),
('Mobile Development', 'Internships in Android/iOS'),
('Networking', 'Internships in computer networks'),
('UI/UX Design', 'Internships in interface design'),
('DevOps', 'Internships in automation and deployment'),
('AI Research', 'Internships in artificial intelligence');
```

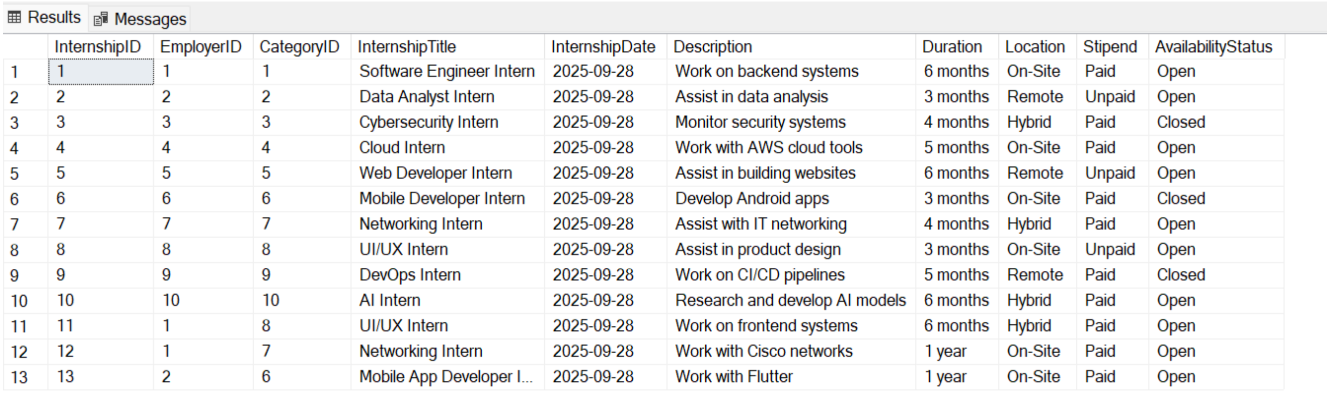


	CategoryID	CategoryName	CategoryDescription
1	1	Software Development	Internships in software engineering and developm...
2	2	Data Science	Internships in analytics and machine learning
3	3	Cybersecurity	Internships in IT security
4	4	Cloud Computing	Internships in AWS, Azure, GCP
5	5	Web Development	Frontend and backend internships
6	6	Mobile Development	Internships in Android/iOS
7	7	Networking	Internships in computer networks
8	8	UI/UX Design	Internships in interface design
9	9	DevOps	Internships in automation and deployment
10	10	AI Research	Internships in artificial intelligence

Internship Table

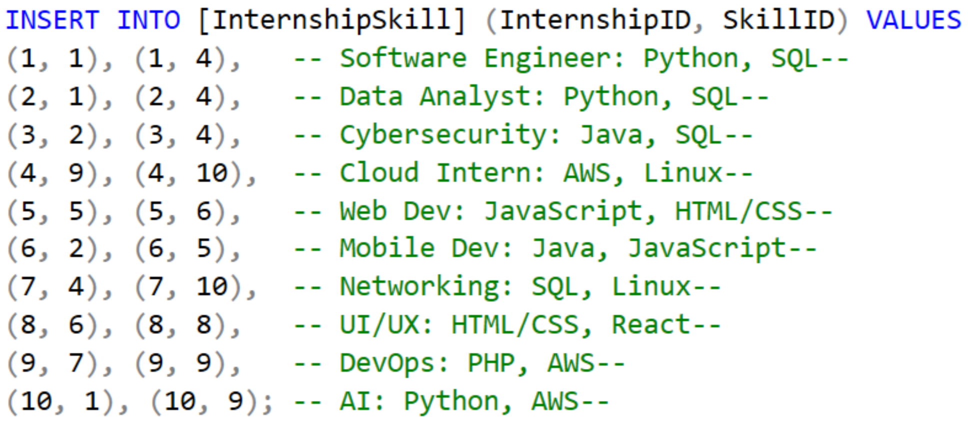


```
INSERT INTO [Internship] (EmployerID, CategoryID, InternshipTitle, Description, Duration, Location, Stipend, AvailabilityStatus) VALUES
(1, 1, 'Software Engineer Intern', 'Work on backend systems', '6 months', 'On-Site', 'Paid', 'Open'),
(2, 2, 'Data Analyst Intern', 'Assist in data analysis', '3 months', 'Remote', 'Unpaid', 'Open'),
(3, 3, 'Cybersecurity Intern', 'Monitor security systems', '4 months', 'Hybrid', 'Paid', 'Closed'),
(4, 4, 'Cloud Intern', 'Work with AWS cloud tools', '5 months', 'On-Site', 'Paid', 'Open'),
(5, 5, 'Web Developer Intern', 'Assist in building websites', '6 months', 'Remote', 'Unpaid', 'Open'),
(6, 6, 'Mobile Developer Intern', 'Develop Android apps', '3 months', 'On-Site', 'Paid', 'Closed'),
(7, 7, 'Networking Intern', 'Assist with IT networking', '4 months', 'Hybrid', 'Paid', 'Open'),
(8, 8, 'UI/UX Intern', 'Assist in product design', '3 months', 'On-Site', 'Unpaid', 'Open'),
(9, 9, 'DevOps Intern', 'Work on CI/CD pipelines', '5 months', 'Remote', 'Paid', 'Closed'),
(10, 10, 'AI Intern', 'Research and develop AI models', '6 months', 'Hybrid', 'Paid', 'Open'),
(1, 8, 'UI/UX Intern', 'Work on frontend systems', '6 months', 'Hybrid', 'Paid', 'Open'),
(1, 7, 'Networking Intern', 'Work with Cisco networks', '1 year', 'On-Site', 'Paid', 'Open'),
(2, 6, 'Mobile App Developer Intern', 'Work with Flutter', '1 year', 'On-Site', 'Paid', 'Open');
```

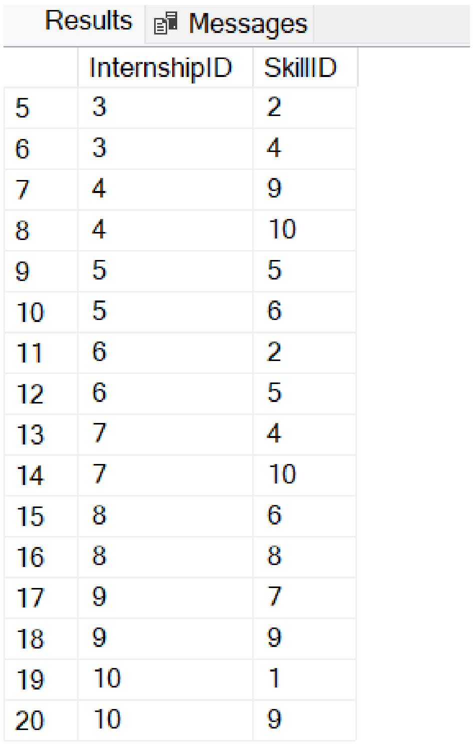


	InternshipID	EmployerID	CategoryID	InternshipTitle	InternshipDate	Description	Duration	Location	Stipend	AvailabilityStatus
1	1	1	1	Software Engineer Intern	2025-09-28	Work on backend systems	6 months	On-Site	Paid	Open
2	2	2	2	Data Analyst Intern	2025-09-28	Assist in data analysis	3 months	Remote	Unpaid	Open
3	3	3	3	Cybersecurity Intern	2025-09-28	Monitor security systems	4 months	Hybrid	Paid	Closed
4	4	4	4	Cloud Intern	2025-09-28	Work with AWS cloud tools	5 months	On-Site	Paid	Open
5	5	5	5	Web Developer Intern	2025-09-28	Assist in building websites	6 months	Remote	Unpaid	Open
6	6	6	6	Mobile Developer Intern	2025-09-28	Develop Android apps	3 months	On-Site	Paid	Closed
7	7	7	7	Networking Intern	2025-09-28	Assist with IT networking	4 months	Hybrid	Paid	Open
8	8	8	8	UI/UX Intern	2025-09-28	Assist in product design	3 months	On-Site	Unpaid	Open
9	9	9	9	DevOps Intern	2025-09-28	Work on CI/CD pipelines	5 months	Remote	Paid	Closed
10	10	10	10	AI Intern	2025-09-28	Research and develop AI models	6 months	Hybrid	Paid	Open
11	11	1	8	UI/UX Intern	2025-09-28	Work on frontend systems	6 months	Hybrid	Paid	Open
12	12	1	7	Networking Intern	2025-09-28	Work with Cisco networks	1 year	On-Site	Paid	Open
13	13	2	6	Mobile App Developer I...	2025-09-28	Work with Flutter	1 year	On-Site	Paid	Open

InternshipSkill Table



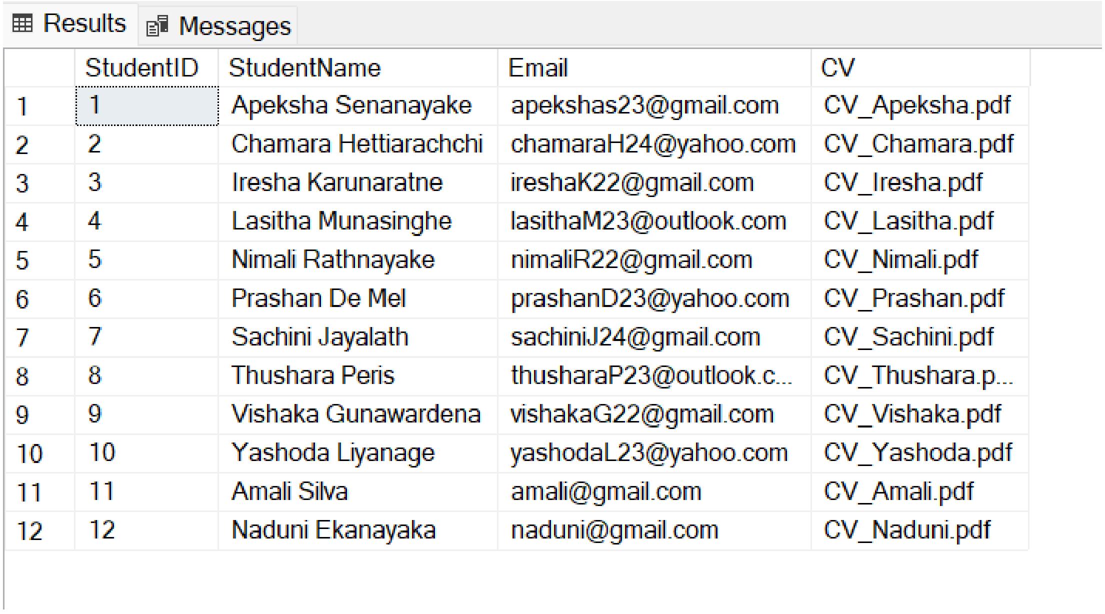
```
INSERT INTO [InternshipSkill] (InternshipID, SkillID) VALUES
(1, 1), (1, 4),   -- Software Engineer: Python, SQL--
(2, 1), (2, 4),   -- Data Analyst: Python, SQL--
(3, 2), (3, 4),   -- Cybersecurity: Java, SQL--
(4, 9), (4, 10),  -- Cloud Intern: AWS, Linux--
(5, 5), (5, 6),   -- Web Dev: JavaScript, HTML/CSS--
(6, 2), (6, 5),   -- Mobile Dev: Java, JavaScript--
(7, 4), (7, 10),  -- Networking: SQL, Linux--
(8, 6), (8, 8),   -- UI/UX: HTML/CSS, React--
(9, 7), (9, 9),   -- DevOps: PHP, AWS--
(10, 1), (10, 9); -- AI: Python, AWS--
```



	InternshipID	SkillID
5	3	2
6	3	4
7	4	9
8	4	10
9	5	5
10	5	6
11	6	2
12	6	5
13	7	4
14	7	10
15	8	6
16	8	8
17	9	7
18	9	9
19	10	1
20	10	9

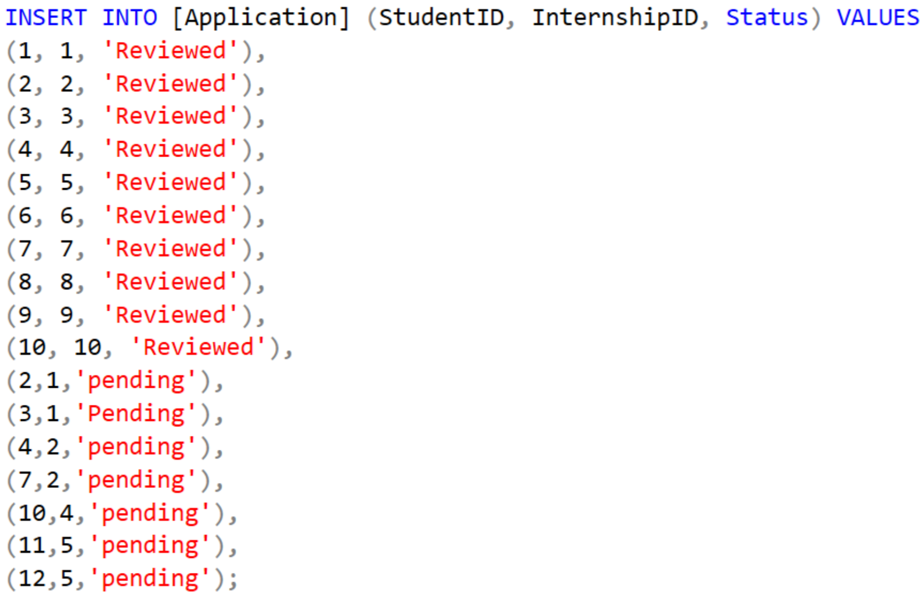
Student Table

```sql
INSERT INTO [Student] (StudentName, Email, CV) VALUES
('Apeksha Senanayake', 'apekshas23@gmail.com', 'CV_Apeksha.pdf'),
('Chamara Hettiarachchi', 'chamaraH24@yahoo.com', 'CV_Chamara.pdf'),
('Iresha Karunaratne', 'ireshaK22@gmail.com', 'CV_Iresha.pdf'),
('Lasitha Munasinghe', 'lasithaM23@outlook.com', 'CV_Lasitha.pdf'),
('Nimali Rathnayake', 'nimaliR22@gmail.com', 'CV_Nimali.pdf'),
('Prashan De Mel', 'prashanD23@yahoo.com', 'CV_Prashan.pdf'),
('Sachini Jayalath', 'sachiniJ24@gmail.com', 'CV_Sachini.pdf'),
('Thushara Peris', 'thusharaP23@outlook.com', 'CV_Thushara.pdf'),
('Vishaka Gunawardena', 'vishakaG22@gmail.com', 'CV_Vishaka.pdf'),
('Yashoda Liyanage', 'yashodaL23@yahoo.com', 'CV_Yashoda.pdf'),
('Amali Silva', 'amali@gmail.com', 'CV_Amali.pdf'),
('Naduni Ekanayaka', 'naduni@gmail.com', 'CV_Naduni.pdf');
```

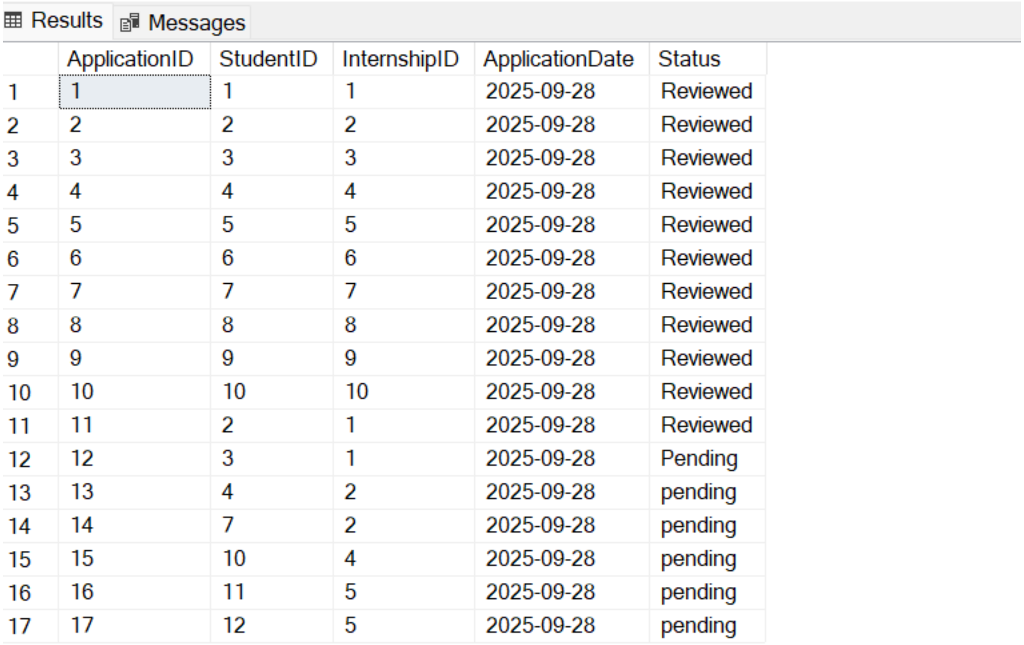


	StudentID	StudentName	Email	CV
1	1	Apeksha Senanayake	apekshas23@gmail.com	CV_Apeksha.pdf
2	2	Chamara Hettiarachchi	chamaraH24@yahoo.com	CV_Chamara.pdf
3	3	Iresha Karunaratne	ireshaK22@gmail.com	CV_Iresha.pdf
4	4	Lasitha Munasinghe	lasithaM23@outlook.com	CV_Lasitha.pdf
5	5	Nimali Rathnayake	nimaliR22@gmail.com	CV_Nimali.pdf
6	6	Prashan De Mel	prashanD23@yahoo.com	CV_Prashan.pdf
7	7	Sachini Jayalath	sachiniJ24@gmail.com	CV_Sachini.pdf
8	8	Thushara Peris	thusharaP23@outlook.c...	CV_Thushara.p...
9	9	Vishaka Gunawardena	vishakaG22@gmail.com	CV_Vishaka.pdf
10	10	Yashoda Liyanage	yashodaL23@yahoo.com	CV_Yashoda.pdf
11	11	Amali Silva	amali@gmail.com	CV_Amali.pdf
12	12	Naduni Ekanayaka	naduni@gmail.com	CV_Naduni.pdf

Application Table

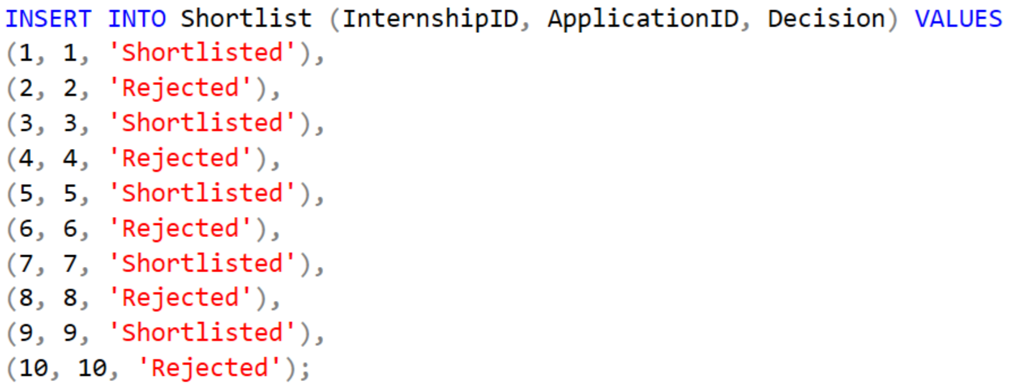


```
INSERT INTO [Application] (StudentID, InternshipID, Status) VALUES
(1, 1, 'Reviewed'),
(2, 2, 'Reviewed'),
(3, 3, 'Reviewed'),
(4, 4, 'Reviewed'),
(5, 5, 'Reviewed'),
(6, 6, 'Reviewed'),
(7, 7, 'Reviewed'),
(8, 8, 'Reviewed'),
(9, 9, 'Reviewed'),
(10, 10, 'Reviewed'),
(2,1,'pending'),
(3,1,'Pending'),
(4,2,'pending'),
(7,2,'pending'),
(10,4,'pending'),
(11,5,'pending'),
(12,5,'pending');
```

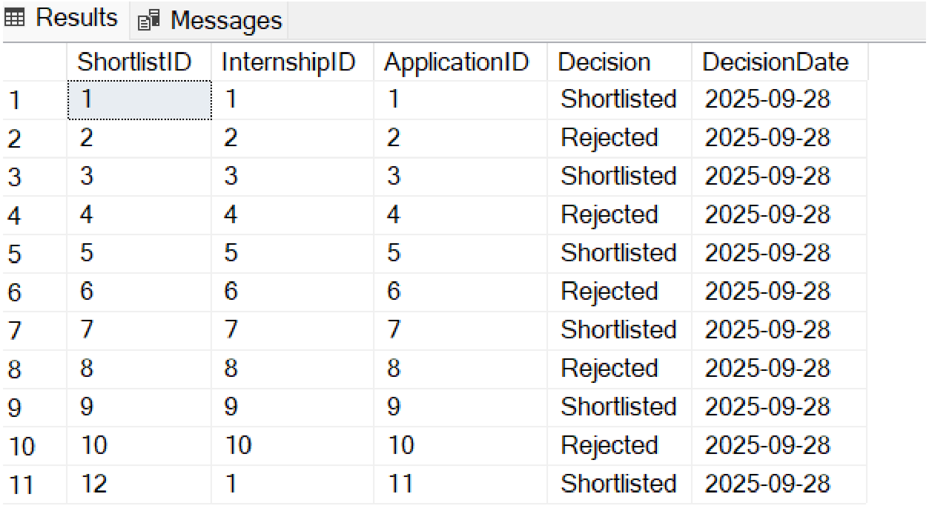


	ApplicationID	StudentID	InternshipID	ApplicationDate	Status
1	1	1	1	2025-09-28	Reviewed
2	2	2	2	2025-09-28	Reviewed
3	3	3	3	2025-09-28	Reviewed
4	4	4	4	2025-09-28	Reviewed
5	5	5	5	2025-09-28	Reviewed
6	6	6	6	2025-09-28	Reviewed
7	7	7	7	2025-09-28	Reviewed
8	8	8	8	2025-09-28	Reviewed
9	9	9	9	2025-09-28	Reviewed
10	10	10	10	2025-09-28	Reviewed
11	11	2	1	2025-09-28	Reviewed
12	12	3	1	2025-09-28	Pending
13	13	4	2	2025-09-28	pending
14	14	7	2	2025-09-28	pending
15	15	10	4	2025-09-28	pending
16	16	11	5	2025-09-28	pending
17	17	12	5	2025-09-28	pending

Shortlist Table



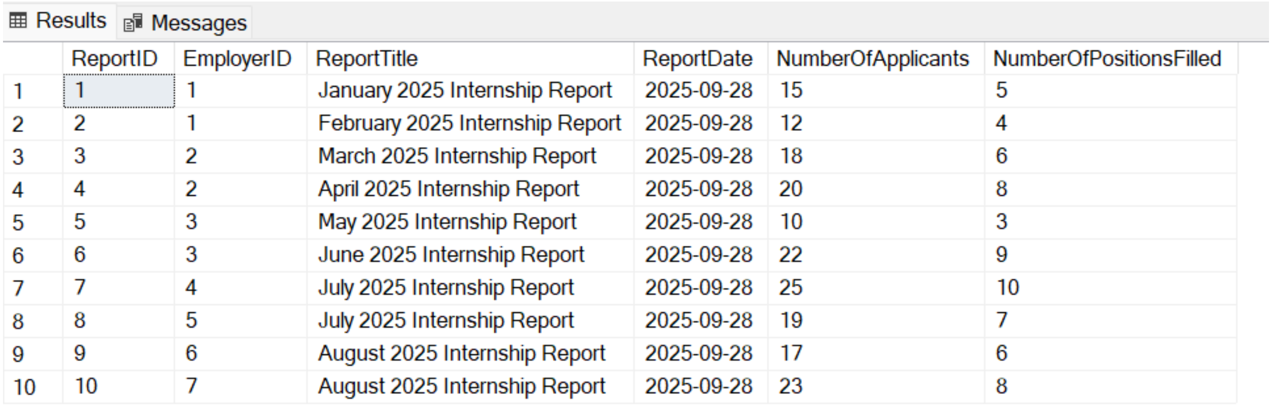
```
INSERT INTO Shortlist (InternshipID, ApplicationID, Decision) VALUES
(1, 1, 'Shortlisted'),
(2, 2, 'Rejected'),
(3, 3, 'Shortlisted'),
(4, 4, 'Rejected'),
(5, 5, 'Shortlisted'),
(6, 6, 'Rejected'),
(7, 7, 'Shortlisted'),
(8, 8, 'Rejected'),
(9, 9, 'Shortlisted'),
(10, 10, 'Rejected');
```



	ShortlistID	InternshipID	ApplicationID	Decision	DecisionDate
1	1	1	1	Shortlisted	2025-09-28
2	2	2	2	Rejected	2025-09-28
3	3	3	3	Shortlisted	2025-09-28
4	4	4	4	Rejected	2025-09-28
5	5	5	5	Shortlisted	2025-09-28
6	6	6	6	Rejected	2025-09-28
7	7	7	7	Shortlisted	2025-09-28
8	8	8	8	Rejected	2025-09-28
9	9	9	9	Shortlisted	2025-09-28
10	10	10	10	Rejected	2025-09-28
11	12	1	11	Shortlisted	2025-09-28

Report Table

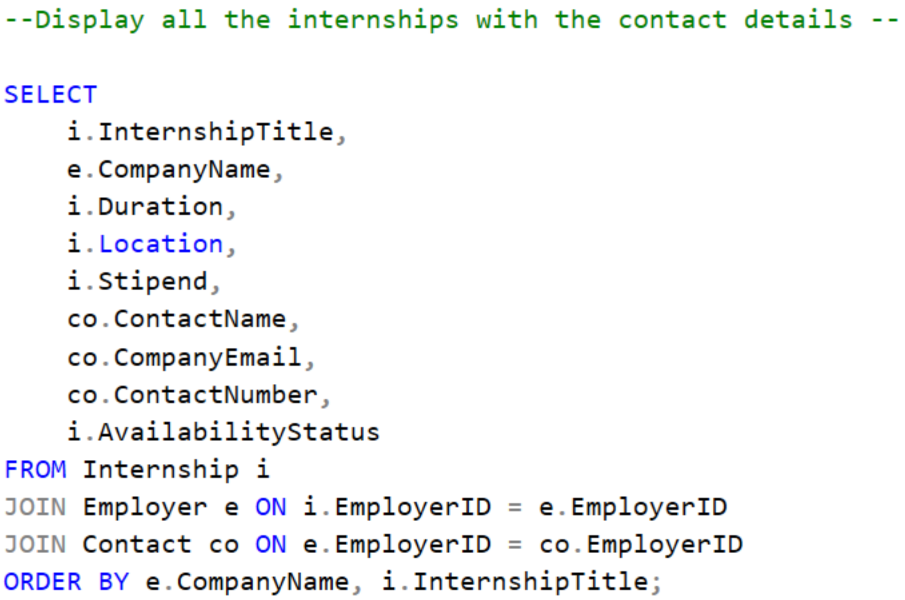
```
INSERT INTO [Report] (EmployerID, ReportTitle, NumberOfApplicants, NumberOfPositionsFilled) VALUES
(1, 'January 2025 Internship Report', 15, 5),
(1, 'February 2025 Internship Report', 12, 4),
(2, 'March 2025 Internship Report', 18, 6),
(2, 'April 2025 Internship Report', 20, 8),
(3, 'May 2025 Internship Report', 10, 3),
(3, 'June 2025 Internship Report', 22, 9),
(4, 'July 2025 Internship Report', 25, 10),
(5, 'July 2025 Internship Report', 19, 7),
(6, 'August 2025 Internship Report', 17, 6),
(7, 'August 2025 Internship Report', 23, 8);
```



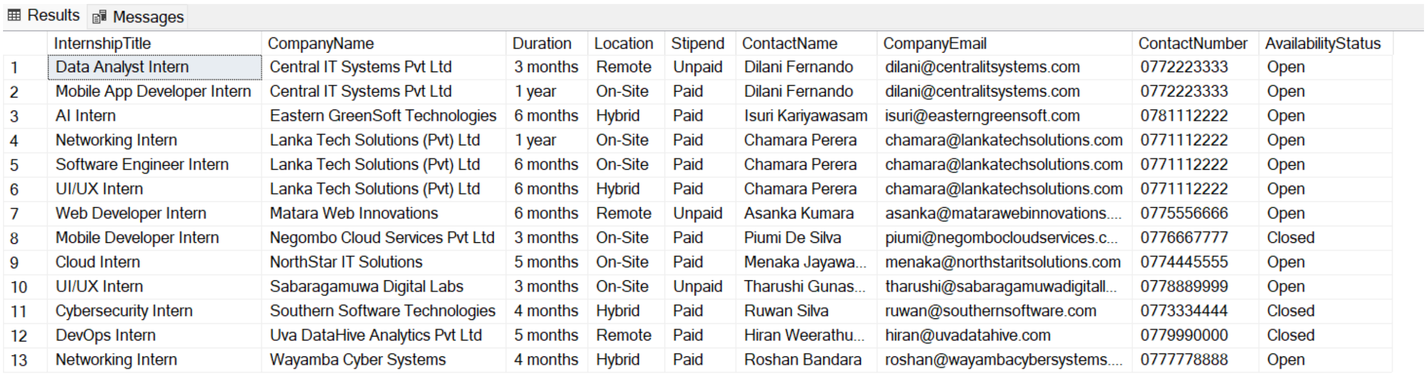
	ReportID	EmployerID	ReportTitle	ReportDate	NumberOfApplicants	NumberOfPositionsFilled
1	1	1	January 2025 Internship Report	2025-09-28	15	5
2	2	1	February 2025 Internship Report	2025-09-28	12	4
3	3	2	March 2025 Internship Report	2025-09-28	18	6
4	4	2	April 2025 Internship Report	2025-09-28	20	8
5	5	3	May 2025 Internship Report	2025-09-28	10	3
6	6	3	June 2025 Internship Report	2025-09-28	22	9
7	7	4	July 2025 Internship Report	2025-09-28	25	10
8	8	5	July 2025 Internship Report	2025-09-28	19	7
9	9	6	August 2025 Internship Report	2025-09-28	17	6
10	10	7	August 2025 Internship Report	2025-09-28	23	8

Query Reports

1. Display all the internships with the contact details

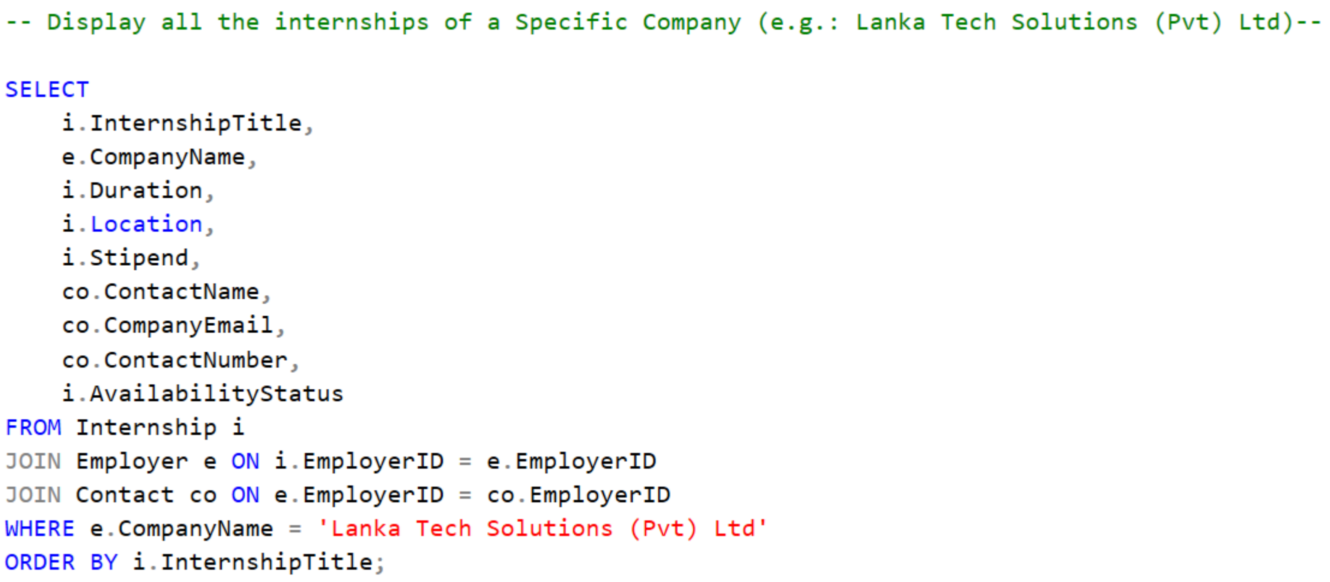


```sql
--Display all the internships with the contact details --

SELECT
    i.InternshipTitle,
    e.CompanyName,
    i.Duration,
    i.Location,
    i.Stipend,
    co.ContactName,
    co.CompanyEmail,
    co.ContactNumber,
    i.AvailabilityStatus
FROM Internship i
JOIN Employer e ON i.EmployerID = e.EmployerID
JOIN Contact co ON e.EmployerID = co.EmployerID
ORDER BY e.CompanyName, i.InternshipTitle;
```

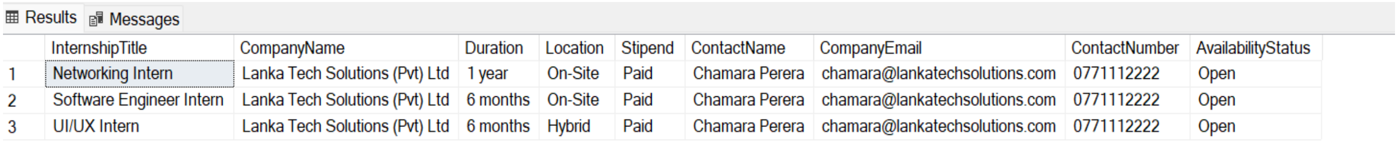


	InternshipTitle	CompanyName	Duration	Location	Stipend	ContactName	CompanyEmail	ContactNumber	AvailabilityStatus
1	Data Analyst Intern	Central IT Systems Pvt Ltd	3 months	Remote	Unpaid	Dilani Fernando	dilani@centralitsystems.com	0772223333	Open
2	Mobile App Developer Intern	Central IT Systems Pvt Ltd	1 year	On-Site	Paid	Dilani Fernando	dilani@centralitsystems.com	0772223333	Open
3	AI Intern	Eastern GreenSoft Technologies	6 months	Hybrid	Paid	Isuri Kariyawasam	isuri@easterngreensoft.com	0781112222	Open
4	Networking Intern	Lanka Tech Solutions (Pvt) Ltd	1 year	On-Site	Paid	Chamara Perera	chamara@lankatechsolutions.com	0771112222	Open
5	Software Engineer Intern	Lanka Tech Solutions (Pvt) Ltd	6 months	On-Site	Paid	Chamara Perera	chamara@lankatechsolutions.com	0771112222	Open
6	UI/UX Intern	Lanka Tech Solutions (Pvt) Ltd	6 months	Hybrid	Paid	Chamara Perera	chamara@lankatechsolutions.com	0771112222	Open
7	Web Developer Intern	Matara Web Innovations	6 months	Remote	Unpaid	Asanka Kumara	asanka@matarawebinnovations....	0775556666	Open
8	Mobile Developer Intern	Negombo Cloud Services Pvt Ltd	3 months	On-Site	Paid	Piumi De Silva	piumi@negombocloudservices.c...	0776667777	Closed
9	Cloud Intern	NorthStar IT Solutions	5 months	On-Site	Paid	Menaka Jayawa...	menaka@northstaritsolutions.com	0774445555	Open
10	UI/UX Intern	Sabaragamuwa Digital Labs	3 months	On-Site	Unpaid	Tharushi Gunas...	tharushi@sabaragamuwadigitall...	0778889999	Open
11	Cybersecurity Intern	Southern Software Technologies	4 months	Hybrid	Paid	Ruwan Silva	ruwan@southernsoftware.com	0773334444	Closed
12	DevOps Intern	Uva DataHive Analytics Pvt Ltd	5 months	Remote	Paid	Hiran Weerathu...	hiran@uvadatahive.com	0779990000	Closed
13	Networking Intern	Wayamba Cyber Systems	4 months	Hybrid	Paid	Roshan Bandara	roshan@wayambacybersystems....	0777778888	Open

2. Display all the internships of a Specific Company (e.g.: Lanka Tech Solutions (Pvt) Ltd)

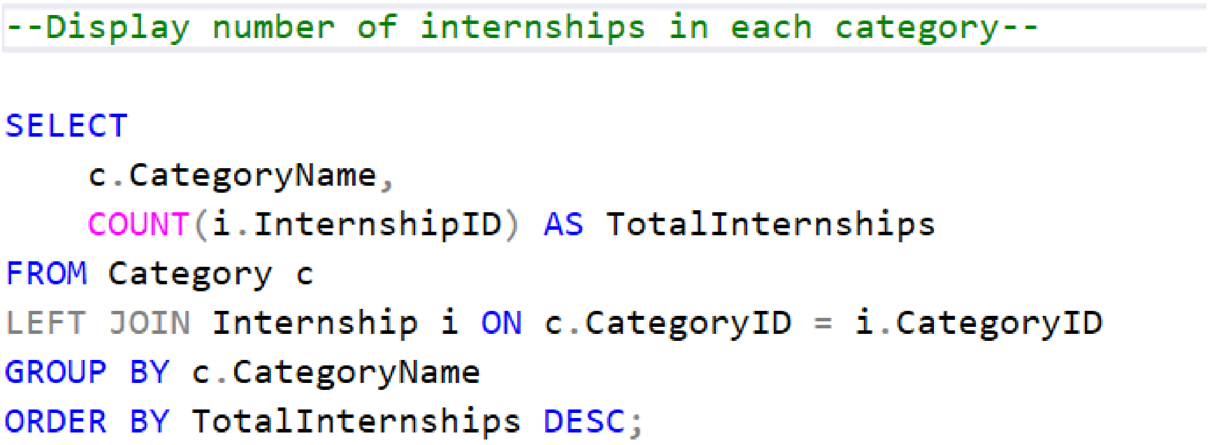


```sql
-- Display all the internships of a Specific Company (e.g.: Lanka Tech Solutions (Pvt) Ltd)--

SELECT
    i.InternshipTitle,
    e.CompanyName,
    i.Duration,
    i.Location,
    i.Stipend,
    co.ContactName,
    co.CompanyEmail,
    co.ContactNumber,
    i.AvailabilityStatus
FROM Internship i
JOIN Employer e ON i.EmployerID = e.EmployerID
JOIN Contact co ON e.EmployerID = co.EmployerID
WHERE e.CompanyName = 'Lanka Tech Solutions (Pvt) Ltd'
ORDER BY i.InternshipTitle;
```

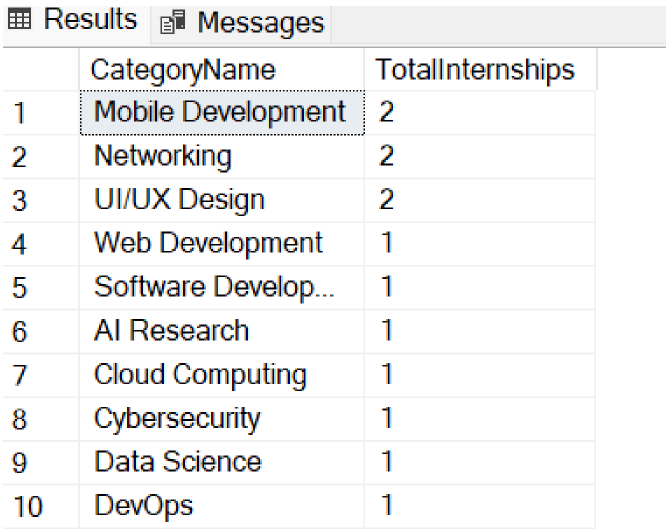


	InternshipTitle	CompanyName	Duration	Location	Stipend	ContactName	CompanyEmail	ContactNumber	AvailabilityStatus
1	Networking Intern	Lanka Tech Solutions (Pvt) Ltd	1 year	On-Site	Paid	Chamara Perera	chamara@lankatechsolutions.com	0771112222	Open
2	Software Engineer Intern	Lanka Tech Solutions (Pvt) Ltd	6 months	On-Site	Paid	Chamara Perera	chamara@lankatechsolutions.com	0771112222	Open
3	UI/UX Intern	Lanka Tech Solutions (Pvt) Ltd	6 months	Hybrid	Paid	Chamara Perera	chamara@lankatechsolutions.com	0771112222	Open

3. Display number of internships in each category

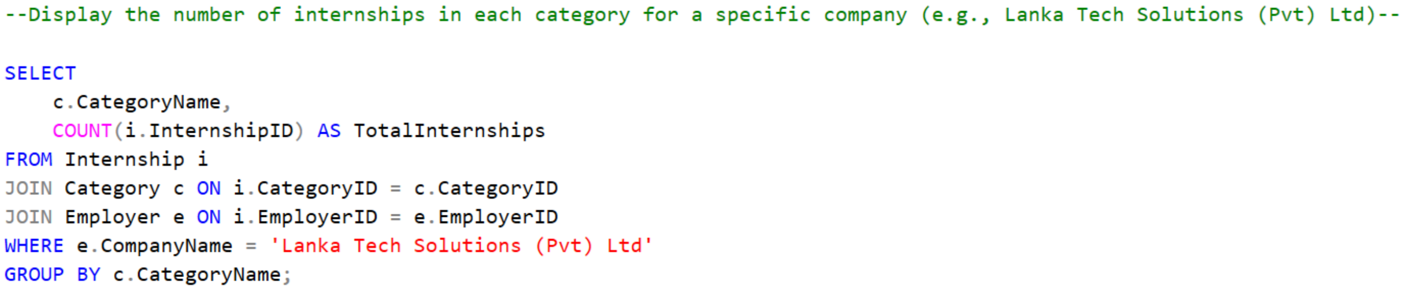


```sql
--Display number of internships in each category--

SELECT
    c.CategoryName,
    COUNT(i.InternshipID) AS TotalInternships
FROM Category c
LEFT JOIN Internship i ON c.CategoryID = i.CategoryID
GROUP BY c.CategoryName
ORDER BY TotalInternships DESC;
```

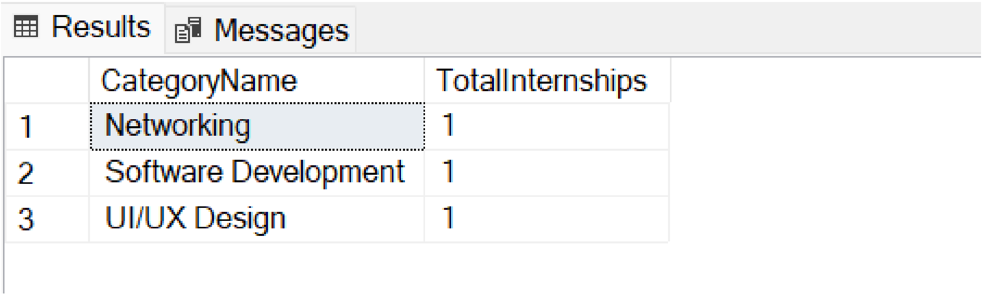


	CategoryName	TotalInternships
1	Mobile Development	2
2	Networking	2
3	UI/UX Design	2
4	Web Development	1
5	Software Develop...	1
6	AI Research	1
7	Cloud Computing	1
8	Cybersecurity	1
9	Data Science	1
10	DevOps	1

4. Display the number of internships in each category for a specific company (e.g., Lanka Tech Solutions (Pvt) Ltd)

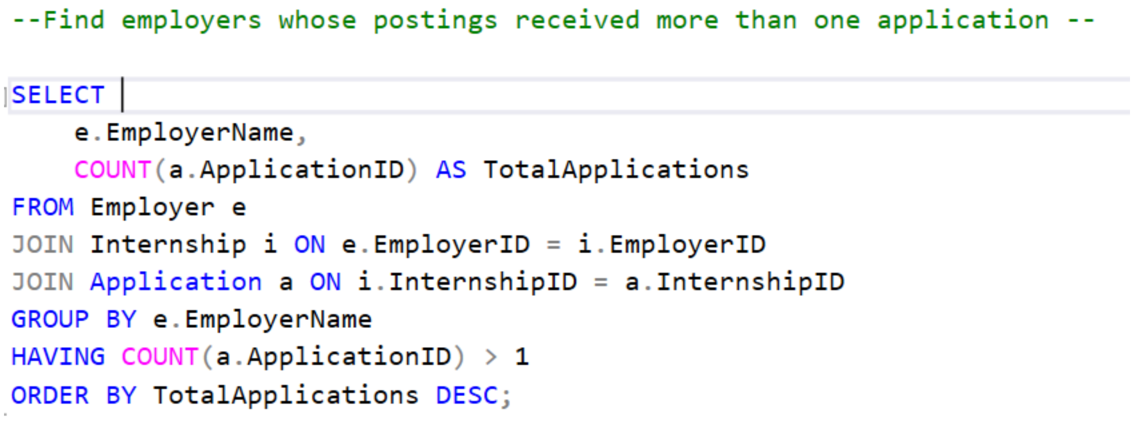


```sql
--Display the number of internships in each category for a specific company (e.g., Lanka Tech Solutions (Pvt) Ltd)--

SELECT
    c.CategoryName,
    COUNT(i.InternshipID) AS TotalInternships
FROM Internship i
JOIN Category c ON i.CategoryID = c.CategoryID
JOIN Employer e ON i.EmployerID = e.EmployerID
WHERE e.CompanyName = 'Lanka Tech Solutions (Pvt) Ltd'
GROUP BY c.CategoryName;
```

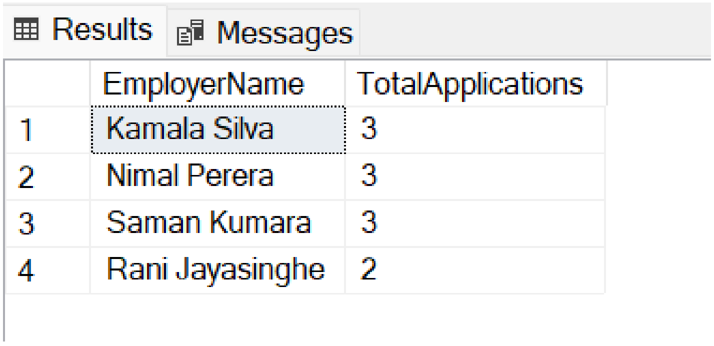


	CategoryName	TotalInternships
1	Networking	1
2	Software Development	1
3	UI/UX Design	1

5. Find employers whose postings received more than one application

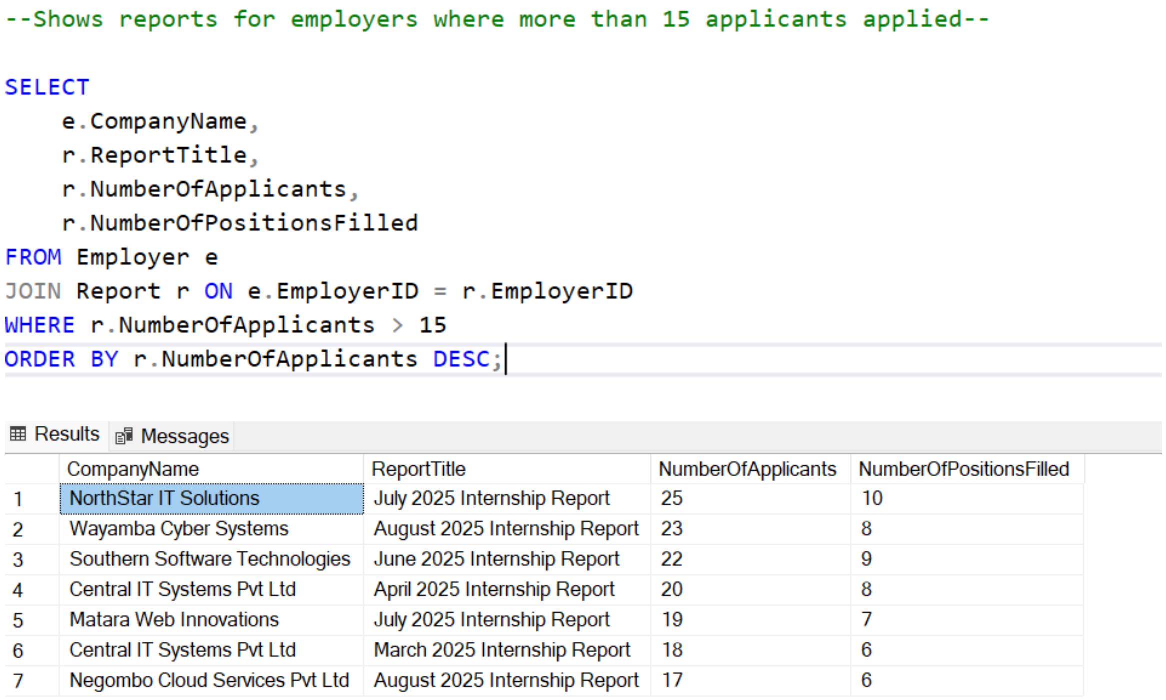


```
--Find employers whose postings received more than one application --

SELECT
    e.EmployerName,
    COUNT(a.ApplicationID) AS TotalApplications
FROM Employer e
JOIN Internship i ON e.EmployerID = i.EmployerID
JOIN Application a ON i.InternshipID = a.InternshipID
GROUP BY e.EmployerName
HAVING COUNT(a.ApplicationID) > 1
ORDER BY TotalApplications DESC;
```



	EmployerName	TotalApplications
1	Kamala Silva	3
2	Nimal Perera	3
3	Saman Kumara	3
4	Rani Jayasinghe	2

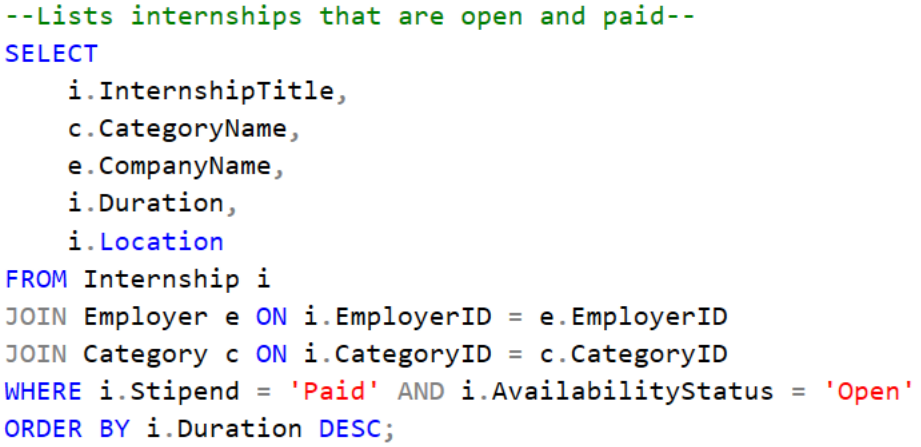
6. Shows reports for employers where more than 15 applicants applied



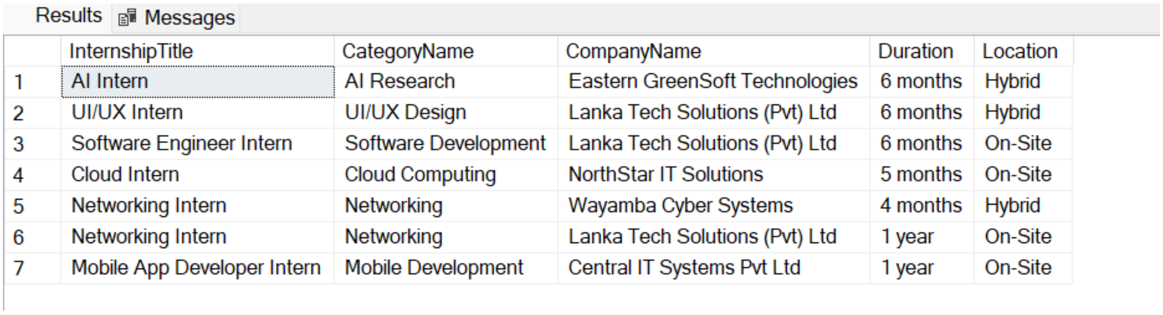
```
--Shows reports for employers where more than 15 applicants applied--

SELECT
    e.CompanyName,
    r.ReportTitle,
    r.NumberOfApplicants,
    r.NumberOfPositionsFilled
FROM Employer e
JOIN Report r ON e.EmployerID = r.EmployerID
WHERE r.NumberOfApplicants > 15
ORDER BY r.NumberOfApplicants DESC;
```

	CompanyName	ReportTitle	NumberOfApplicants	NumberOfPositionsFilled
1	NorthStar IT Solutions	July 2025 Internship Report	25	10
2	Wayamba Cyber Systems	August 2025 Internship Report	23	8
3	Southern Software Technologies	June 2025 Internship Report	22	9
4	Central IT Systems Pvt Ltd	April 2025 Internship Report	20	8
5	Matara Web Innovations	July 2025 Internship Report	19	7
6	Central IT Systems Pvt Ltd	March 2025 Internship Report	18	6
7	Negombo Cloud Services Pvt Ltd	August 2025 Internship Report	17	6

7. Lists internships that are open and paid

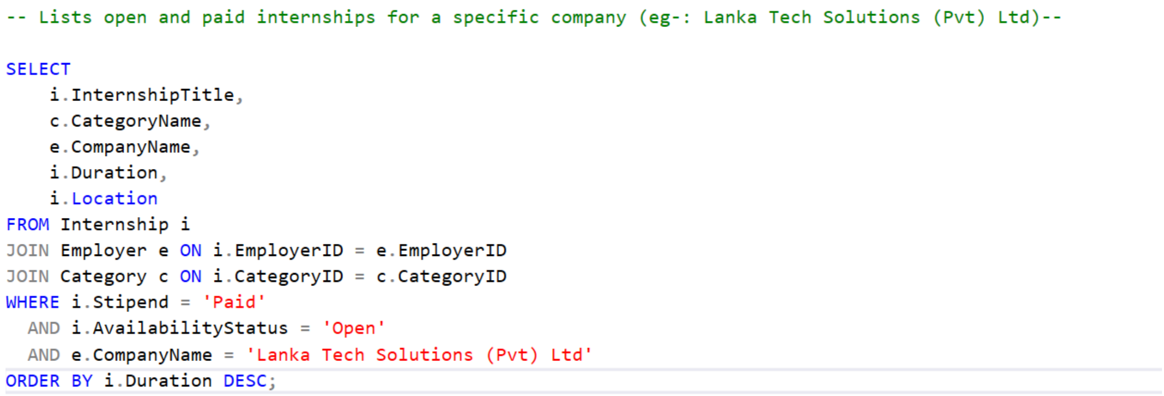


```
--Lists internships that are open and paid--
SELECT
    i.InternshipTitle,
    c.CategoryName,
    e.CompanyName,
    i.Duration,
    i.Location
FROM Internship i
JOIN Employer e ON i.EmployerID = e.EmployerID
JOIN Category c ON i.CategoryID = c.CategoryID
WHERE i.Stipend = 'Paid' AND i.AvailabilityStatus = 'Open'
ORDER BY i.Duration DESC;
```

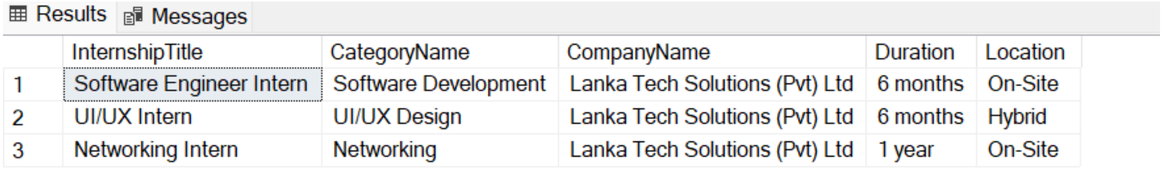


	InternshipTitle	CategoryName	CompanyName	Duration	Location
1	AI Intern	AI Research	Eastern GreenSoft Technologies	6 months	Hybrid
2	UI/UX Intern	UI/UX Design	Lanka Tech Solutions (Pvt) Ltd	6 months	Hybrid
3	Software Engineer Intern	Software Development	Lanka Tech Solutions (Pvt) Ltd	6 months	On-Site
4	Cloud Intern	Cloud Computing	NorthStar IT Solutions	5 months	On-Site
5	Networking Intern	Networking	Wayamba Cyber Systems	4 months	Hybrid
6	Networking Intern	Networking	Lanka Tech Solutions (Pvt) Ltd	1 year	On-Site
7	Mobile App Developer Intern	Mobile Development	Central IT Systems Pvt Ltd	1 year	On-Site

8. Lists open and paid internships for a specific company (eg-: Lanka Tech Solutions (Pvt) Ltd)



```
-- Lists open and paid internships for a specific company (eg-: Lanka Tech Solutions (Pvt) Ltd)--

SELECT
    i.InternshipTitle,
    c.CategoryName,
    e.CompanyName,
    i.Duration,
    i.Location
FROM Internship i
JOIN Employer e ON i.EmployerID = e.EmployerID
JOIN Category c ON i.CategoryID = c.CategoryID
WHERE i.Stipend = 'Paid'
  AND i.AvailabilityStatus = 'Open'
  AND e.CompanyName = 'Lanka Tech Solutions (Pvt) Ltd'
ORDER BY i.Duration DESC;
```



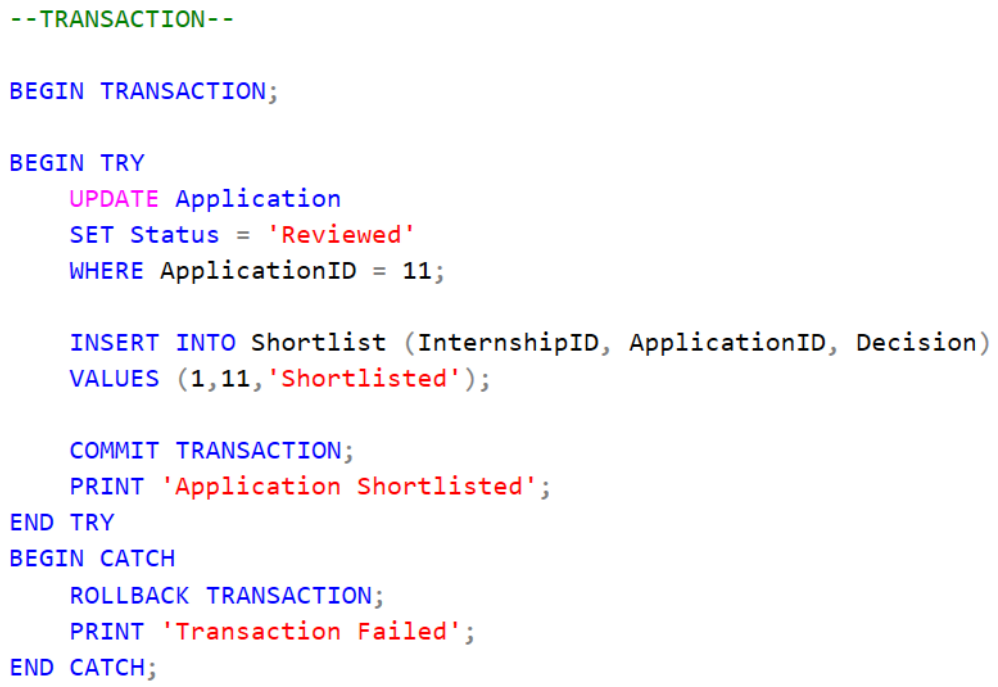
	InternshipTitle	CategoryName	CompanyName	Duration	Location
1	Software Engineer Intern	Software Development	Lanka Tech Solutions (Pvt) Ltd	6 months	On-Site
2	UI/UX Intern	UI/UX Design	Lanka Tech Solutions (Pvt) Ltd	6 months	Hybrid
3	Networking Intern	Networking	Lanka Tech Solutions (Pvt) Ltd	1 year	On-Site

Transaction

Transaction for shortlisting applications

This ensures that when an employer shortlists a student for an internship, both the application status and the shortlist entry are both updated simultaneously as one atomic operation.

- First, the Application table is updated to mark the application as Reviewed.
- Next, a corresponding record is inserted in the Shortlist table with the employer's selection.
- If both operations are successfully completed, the transaction is committed, the changes will be saved.
- If any error is faced due to wrong IDs or due to a constraint error the transaction is rolled back, and no changes will be made.



```
--TRANSACTION--

BEGIN TRANSACTION;

BEGIN TRY
    UPDATE Application
    SET Status = 'Reviewed'
    WHERE ApplicationID = 11;

    INSERT INTO Shortlist (InternshipID, ApplicationID, Decision)
    VALUES (1,11,'Shortlisted');

    COMMIT TRANSACTION;
    PRINT 'Application Shortlisted';
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Transaction Failed';
END CATCH;
```

References

Microsoft (n.d.) *Install SQL Server Management Studio*. Available at: https://learn.microsoft.com/en-us/ssms/install/install (Accessed: 12 August 2025).

Oracle (2023) *MySQL documentation*. Available at: https://dev.mysql.com/doc/ (Accessed: 26 September 2025).

MongoDB (2023) *MongoDB vs MySQL*. Available at: https://www.mongodb.com/compare/mongodb-mysql (Accessed: 26 September 2025).

W3Schools (n.d.) *SQL Tutorial*. Available at: https://www.w3schools.com/sql/ (Accessed: 29 September 2025).

AWS (n.d.) *What is SQL?*. Available at: https://aws.amazon.com/what-is/sql/ (Accessed: 25 September 2025).

draw.io (n.d.) *Flowchart maker & online diagram software*. Available at: https://draw.io/ (Accessed: 28 September 2025).