

# Star Internship Finder

## **Business Scenario**

A local startup, Star, is trying to help college students obtain internships by providing a service that delivers available internships depending on what the student is looking for. However, they currently do not have a database management system to keep track of this data. They currently keep track of their data using pen and paper and realize that this isn't scalable and is inefficient.

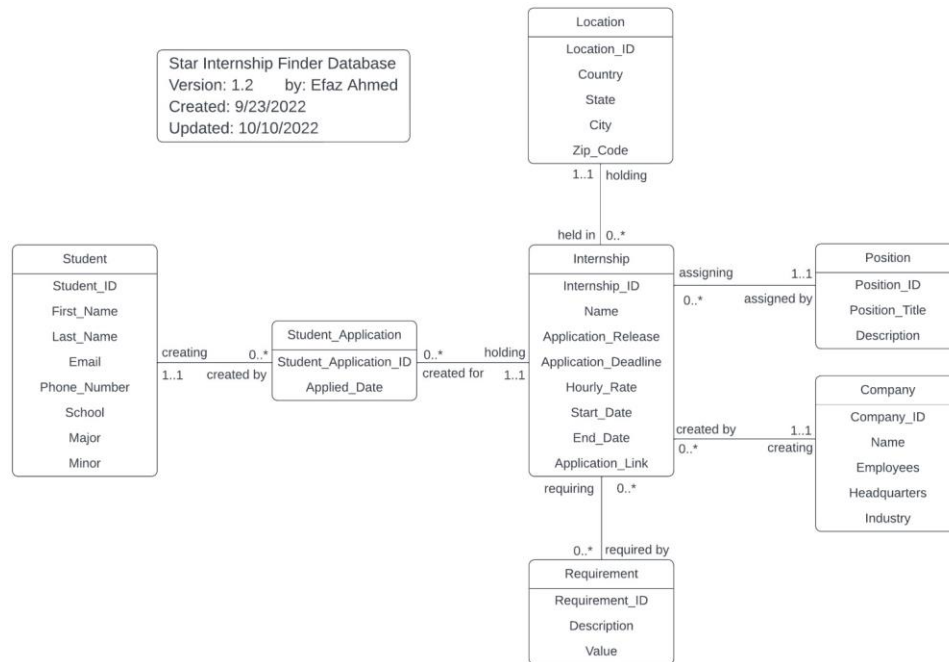
In my business, students can apply for internships depending on many criteria. However, in order to accomplish this, I will need information for solving this issue, such as data regarding each internship's company, position, location, industry, salary, release date, due date, duration, and link. By having information about internship positions available in each company, Star can post a variety of internships on our platform instead of having students themselves search for internships manually.

For the system to work out as planned and for record-keeping purposes, Star needs to keep track of which company releases an internship along with information that goes along with it like positions, release and deadline dates, etc. I will also need to store information about students to keep track of which company they apply to along with their contact information. To accomplish the goal above, Star will have a unique identifier for each student to keep track of what internship they applied to. There will also be a unique identifier for the company to keep track of which company posted an internship on our application. A unique identifier for location would also be needed for students to locate the area they will be interning in if accepted. In addition, a unique identifier is needed for a requirement to allow companies to place onto internships, allowing students who meet those requirements to have a higher chance of being accepted. A unique position identifier allows students to search for specific occupations they are interested in applying to. Lastly, each internship requires a unique identifier to keep track of what company has internship opportunities along with the location, position, and requirement for students to apply to.

### **Entities:**

- Companies
- Positions
- Internships
- Locations
- Students

## E-R Model using UML Notation



## Relationship Sentences

- Entities
- Minimum Cardinality
- Verb Phrase
- Maximum Cardinality

One Internship **must be held in** one Location

One Location **may be holding** one or more Internship

One Internship **must be created by** one Company

One Company **may be creating** one or more Internship

One Internship **must be assigning** one Position

One Position **may be assigned by** one or more Internship

One Internship **may be requiring** one or more Requirement

One Requirement **may be required by** one or more Internship

One Internship **may be holding** one or more Student\_Application

One Student\_Application **must be created for** one Internship

One Student **may be creating** one or more Student\_Application

One Student\_Application **must be created by** one Student

## Conversion to Relational Model

• Entities   • **Key**   • **Foreign Key**

**Internship** (Internship\_ID (**Key**), Name, Application\_Release, Application\_Deadline, Hourly\_Rate, Start\_Date, End\_Date, Application\_Link, Location\_ID (**FK**), Company\_ID (**FK**), Position\_ID (**FK**))

**Location** (Location\_ID (**Key**), Country, State, City, Zip\_Code)

**Company** (Company\_ID (**Key**), Name, Employees, Headquarters, Industry)

**Position** (Position\_ID (**Key**), Position\_Title, Description)

**Requirement** (Requirement\_ID (**Key**), Description, Value)

**Student** (Student\_ID (**Key**), First\_Name, Last\_Name, Email, Phone\_Number, School, Major, Minor)

**Student\_Application** (Student\_Application\_ID (**Key**), Applied\_Date, Status, Student\_ID (**FK**), Internship\_ID (**FK**))

**Internship\_Requirement** (Requirement\_ID (**Key**)(**FK**), Internship\_ID (**Key**)(**FK**))

## Normalization

**Internship** (Internship\_ID, Name, Application\_Release, Application\_Deadline, Hourly\_Rate, Start\_Date, End\_Date, Application\_Link, Location\_ID, Company\_ID, Position\_ID, Requirement\_ID)

- Key: Internship\_ID
- FD1: Internship\_ID → Name, Application\_Release, Application\_Deadline, Hourly\_Rate, Start\_Date, End\_Date, Application\_Link, Location\_ID, Company\_ID, Position\_ID, Requirement\_ID
- This relation is in 3NF because:
  - It's in 1NF, 2NF, and there are no transitive dependencies present.
    - Its 1NF due to it being a relation.
    - It's 2NF because all non-key attributes depend on all the key. **Location** (Location\_ID, Country, State, City, Zip\_Code)

• Key: Location\_ID

• FD1: Location\_ID → Country, State, City, Zip\_Code • This relation is in 3NF because:

- It's in 1NF, 2NF, and there are no transitive dependencies present.
  - Its 1NF because it was split from a relation.
  - It's 2NF because all non-key attributes depend on all the key. **Company** (Company\_ID, Name, Employees, Headquarters, Industry)

• Key: Company\_ID

• FD1: Company\_ID → Name, Employees, Headquarters, Industry • This relation is in 3NF because:

- It's in 1NF, 2NF, and there are no transitive dependencies present.
  - Its 1NF due to it being a relation.
  - It's 2NF because all non-key attributes depend on all the key. **Position** (Position\_ID, Position\_Title, Description)

• Key: Position\_ID

• FD1: Position\_ID → Position\_Title, Description • This relation is in 3NF because:

- It's in 1NF, 2NF, and there are no transitive dependencies present.
  - Its 1NF due to it being a relation.
  - It's 2NF because all non-key attributes depend on all the key.

**Requirement** (Requirement\_ID, Description, Value, Internship\_ID)

- Key: Requirement\_ID
- FD1: Requirement\_ID → Description, Value, Internship\_ID • This relation is in 3NF because:
  - It's in 1NF, 2NF, and there are no transitive dependencies present.
    - Its 1NF due to it being a relation.
    - It's 2NF because all non-key attributes depend on all the key.

**Student** (Student\_ID, First\_Name, Last\_Name, Email, Phone\_Number, School, Major, Minor)

- Key: Student\_ID
- FD1: Student\_ID → First\_Name, Last\_Name, Email, Phone\_Number, School, Major, Minor
- This relation is in 3NF because:
  - It's in 1NF, 2NF, and there are no transitive dependencies present.

- Its 1NF due to it being a relation.
- It's 2NF because all non-key attributes depend on all the key.

**Student\_Application** (Student\_Application\_ID, Applied\_Date, Status, Student\_ID, Internship\_ID)

- Key: Student\_Application\_ID
- FD1: Student\_Application\_ID → Applied\_Date, Status, Student\_ID, Internship\_ID ● This relation is in 3NF because:

- It's in 1NF, 2NF, and there are no transitive dependencies present.

- Its 1NF due to it being a relation.
- It's 2NF because all non-key attributes depend on all the key.

**Internship\_Requirement** (Requirement\_ID, Internship\_ID)

- Key: Requirement\_ID, Internship\_ID
- This relation is in 3NF because:
  - It's in 1NF, 2NF, and there are no transitive dependencies present.
- Its 1NF due to it being a relation.
- It's 2NF because all non-key attributes depend on all the key.

# Creating the Database Schema with Structured Query Language (SQL)

## Creating the Tables

```
CREATE TABLE location (  
    Location_ID          AUTOINCREMENT      NOT NULL,  
    Country              VARCHAR (50)       NOT NULL,  
    State                VARCHAR (2),  
  
    City                 VARCHAR (20),  
  
    Zip_Code             VARCHAR (10),  
    PRIMARY KEY (Location_ID)  
);  
  
CREATE TABLE company (  
    Company_ID          AUTOINCREMENT      NOT NULL,  
    Name                VARCHAR (25)       NOT NULL,  
    Employees           NUMBER,  
  
    Headquarters        VARCHAR (30)       NOT NULL,  
    Industry            VARCHAR (20),  
    PRIMARY KEY (Company_ID)  
);  
  
CREATE TABLE position (  
    Position_ID         AUTOINCREMENT      NOT NULL,  
    Position_Title      VARCHAR (50)       NOT NULL,  
    Description         VARCHAR (250)      NOT NULL,  
    PRIMARY KEY (Position_ID)  
);  
  
CREATE TABLE student (  
    Student_ID          AUTOINCREMENT      NOT NULL,  
    First_Name          VARCHAR (20)       NOT NULL,  
    Last_Name           VARCHAR (75)       NOT NULL,  
    Email               VARCHAR (60),  
  
    Phone_Number        VARCHAR (25),  
  
    School              VARCHAR (50)       NOT NULL,  
    Major               VARCHAR (50),  
  
    Minor               VARCHAR (50),  
    PRIMARY KEY (Student_ID)  
);  
  
CREATE TABLE internship_requirement (  
    Internship_ID       NUMBER              NOT NULL,  
    Requirement_ID       NUMBER              NOT NULL,  
    PRIMARY KEY (Requirement_ID, Internship_ID)  
);  
  
CREATE TABLE requirement (  
    Requirement_ID      AUTOINCREMENT      NOT NULL,  
    Category            VARCHAR (30)       NOT NULL,  
    Condition           VARCHAR (30)       NOT NULL,  
    PRIMARY KEY (Requirement_ID)  
);
```

```

CREATE TABLE internship (

    Internship_ID          AUTOINCREMENT      NOT NULL,
    Name                   VARCHAR (100)      NOT NULL,
    Application_Release DATE, Application_Deadline
    DATE,
    Hourly_Rate            NUMBER,
    Start_Date             DATE,
    End_Date               DATE,
    Application_Link        VARCHAR (200)      NOT NULL,
    Location_ID            NUMBER,
    Company_ID             NUMBER              NOT NULL,
    Position_ID            NUMBER              NOT NULL,
    PRIMARY KEY (Internship_ID)

);

```

```

CREATE TABLE student_application (

    Student_Application_ID  AUTOINCREMENT      NOT NULL,
    Applied_Date            DATE,
    Status                  VARCHAR (25)        NOT NULL,
    Student_ID              NUMBER              NOT NULL,
    Internship_ID           NUMBER              NOT NULL,
    PRIMARY KEY (Student_Application_ID)

);

```

### Adding Foreign Keys

```

ALTER TABLE internship      ADD CONSTRAINT fk_location
FOREIGN KEY (Location_ID)    REFERENCES location (Location_ID);

ALTER TABLE internship      ADD CONSTRAINT fk_company
FOREIGN KEY (Company_ID)     REFERENCES company (Company_ID);

ALTER TABLE internship      ADD CONSTRAINT fk_position
FOREIGN KEY (Position_ID)    REFERENCES position (Position_ID);

ALTER TABLE student_application ADD CONSTRAINT fk_student
FOREIGN KEY (Student_ID)     REFERENCES student (Student_ID);

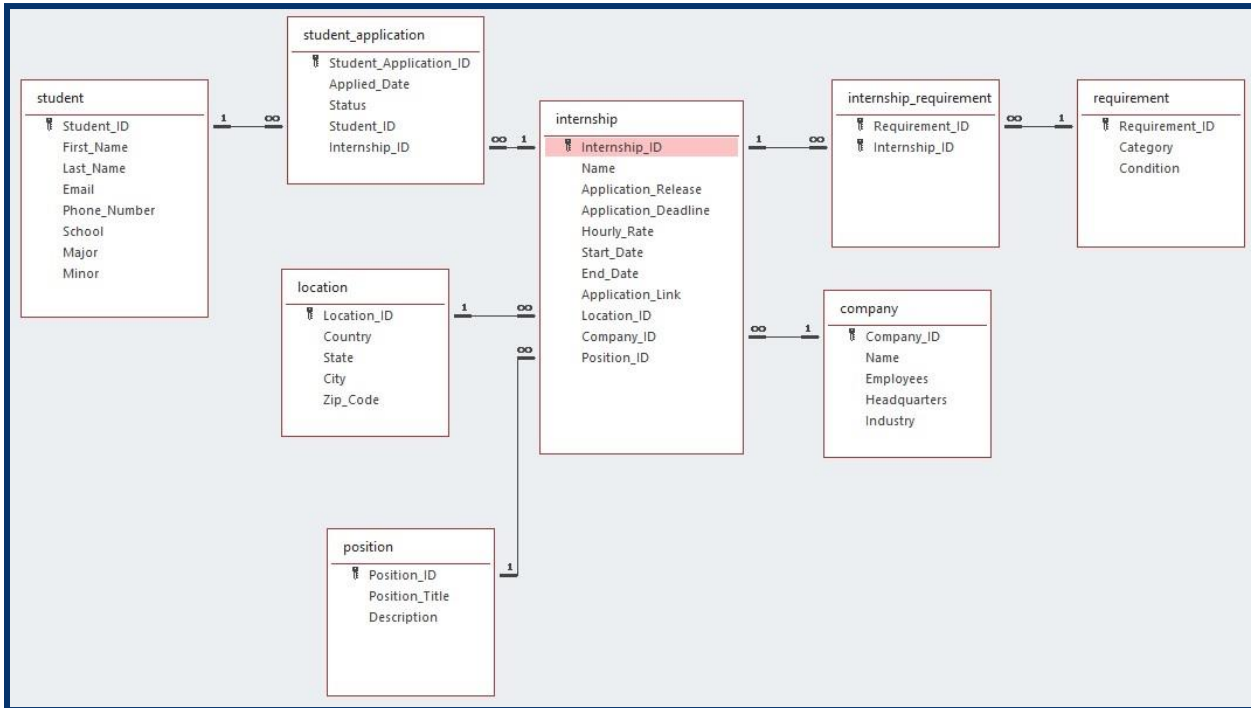
ALTER TABLE student_application ADD CONSTRAINT fk_internship
FOREIGN KEY (Internship_ID)   REFERENCES internship
                              (Internship_ID);

ALTER TABLE internship_requirement ADD CONSTRAINT fk_requirement
FOREIGN KEY (Requirement_ID)    REFERENCES requirement
                              (Requirement_ID);

ALTER TABLE internship_requirement ADD CONSTRAINT fk_internship_req
FOREIGN KEY (Internship_ID)     REFERENCES internship
                              (Internship_ID);

```

## Microsoft Access Relationship View



## Adding Data to the Tables using SQL INSERT Statements

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NY', 'New York', '10001');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NY', 'New York', '10002');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NY', 'New York', '10003');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'GA', 'Columbus', '31904');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NY', 'Maine', '13802');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NY', 'Brooklyn', '11201');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NJ', 'Newark', '07102');
```

```
INSERT INTO location (Country, State, City, Zip_Code)
VALUES ('US', 'NY', 'New York', '10013');
```

```
INSERT INTO company (Name, Employees, Headquarters, Industry)
VALUES ('Netflix', 11300, 'Los Gatos, California', 'Entertainment');
```

```
INSERT INTO company (Name, Employees, Headquarters, Industry)
```



```

VALUES ('Spotify', 6617, 'Stockholm, Sweden', 'Entertainment');

INSERT INTO company (Name, Employees, Headquarters, Industry)
VALUES ('YouTube', 7500, 'San Bruno, California', 'Entertainment');

INSERT INTO company (Name, Employees, Headquarters, Industry)
VALUES ('Etsy', 2402, 'Brooklyn, New York', 'E-commerce');

INSERT INTO company (Name, Employees, Headquarters, Industry)
VALUES ('Yahoo', 10350, 'Sunnyvale, California', 'Software');

INSERT INTO company (Name, Employees, Headquarters, Industry)
VALUES ('Panasonic', 243540, 'Kadoma, Osaka, Japan', 'Electronics');

INSERT INTO company (Name, Employees, Headquarters, Industry)
VALUES ('Two Sigma', 1600, 'SoHo, New York', 'Investment');

INSERT INTO position (Position_Title, Description)
VALUES ('Software Engineer', 'An IT professional who designs, develops, and maintain computer software at a company.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Data Analyst', 'A person who collects, cleans, and interprets data sets in order to answer a question or solve a problem.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Product Manager', 'A person who identifies the customer need and the larger business objectives that a product or feature will fulfill, articulates what success looks like for a product, and rallies a team to turn that vision into reality.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Accountant', 'A person who helps businesses make critical financial decisions by collecting, tracking, and correcting the company's finances.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Cybersecurity', 'A person who safeguards information system assets by identifying and solving potential and actual security problems.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Hardware Engineer', 'A person who researches, designs, develops, and tests computer systems and components such as circuit boards.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Business Analyst', 'A person who analyzes large data sets to identify effective ways of boosting organizational efficiency.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Marketing', 'A person who monitors market trends, creates advertising campaigns, develops pricing strategies and targeting strategies based on demographic data and work with the company to develop more awareness of what they offer.');
```

```

INSERT INTO position (Position_Title, Description)
VALUES ('Quantitative Analyst', 'A person who specializes in "the design, development, and implementation of algorithms and mathematical or statistical models intended to solve complex financial problems.');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Grade Level', 'Freshman');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Grade Level', 'Sophomore');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Grade Level', 'Junior');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Grade Level', 'Senior');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('GPA', '2.0+');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('GPA', '2.5+');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('GPA', '3.0+');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('GPA', '3.5+');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('GPA', '4.0');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Computer Science');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Computer Engineering');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Marketing');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Finance');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Accounting');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Management');
```

```
INSERT INTO requirement (Category, Condition)
VALUES ('Major', 'Communications');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Efaz', 'Ahmed', 'efazahmed@gmail.com', '555-555-5555', 'Baruch College',
'Computer Information Systems', 'Computer Science');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Taohid', 'Shadat', 'taohidshadat@gmail.com', '555-555-5556', 'Baruch
College', 'Computer Information Systems', 'Mathematics');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Colleen', 'Zeng', 'colleenzeng@gmail.com', '555-555-5557', 'Baruch College',
'Computer Information Systems', 'Psychology');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Fawwad', 'Khan', 'fawwadkhan@gmail.com', '555-555-5558', 'Baruch College',
'Computer Information Systems', 'Psychology');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Jeffrey', 'Zheng', 'jeffreyzheng@gmail.com', '555-555-5559', 'Baruch
College', 'Computer Information Systems', 'Communications');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Carlos', 'Silverio', 'carlosilverio@gmail.com', '555-555-5560', 'Baruch
College', 'Statistics', 'Spanish');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Kyle', 'Chen', 'kylechen@gmail.com', '555-555-5561', 'Baruch College',
'Computer Information Systems', 'Natural Sciences');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Dhruv', 'Rajagopala', 'dhruvrajagopala@gmail.com', '555-555-5562', 'Baruch
College', 'Computer Information Systems', 'Communications');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major)
VALUES ('Jason', 'Schwartzman', 'jasonschwartzman@gmail.com', '555-555-5563',
'Binghampton', 'Computer Science');
```

```
INSERT INTO student (First_Name, Last_Name, Email, Phone_Number, School, Major, Minor)
VALUES ('Riya', 'Sawant', 'riyasawant@gmail.com', '555-555-5564', 'Baruch College',
'Marketing Management', 'Psychology');
```

```
INSERT INTO internship (Name, Application_Release, Application_Deadline, Hourly_Rate,
Start_Date, End_Date, Application_Link, Location_ID, Company_ID, Position_ID)
VALUES ('2023 Software Engineering Internship: Backend', '09/15/2022', '11/30/2022', 60,
'06/05/2023', '08/24/2023', 'https://randomjobapplication.com/netflix/1', 2, 1,
1);
```

```
INSERT INTO internship (Name, Application_Release, Application_Deadline, Hourly_Rate,
Start_Date, End_Date, Application_Link, Location_ID, Company_ID, Position_ID)
VALUES ('2023 Summer Financial Data Analyst Intern', '09/08/2022', '11/30/2022', 36,
'06/01/2023', '08/28/2023', 'https://randomjobapplication.com/youtube/1', 3, 3, 2);
```

```
INSERT INTO internship (Name, Application_Release, Application_Deadline, Hourly_Rate,
Start_Date, End_Date, Application_Link, Location_ID, Company_ID, Position_ID)
VALUES ('2023 Technical Program Management Intern', '09/05/2022', '12/30/2022', 55,
'06/02/2023', '08/30/2023', 'https://randomjobapplication.com/youtube/2', 3, 3, 2);
```

```
INSERT INTO internship (Name, Application_Release, Application_Deadline, Hourly_Rate,
Start_Date, End_Date, Application_Link, Location_ID, Company_ID, Position_ID)
VALUES ('Android Software Engineering Intern', '11/26/2022', '1/2/2023', 56,
'05/30/2023', '08/18/2023', 'https://jobs.smartrecruiters.com/Etsy2/743999866391296-android-
software-engineering-i ntern?source=Linkedin', 6, 4, 1);
```

```
INSERT INTO internship (Name, Application_Release, Application_Deadline, Hourly_Rate,
Start_Date, End_Date, Application_Link, Company_ID, Position_ID)
VALUES ('Software Apps Engineerings Intern', '12/10/2023', '1/2/2023', 38,
'06/01/2023', '08/25/2023', 'https://ouryahoo.wd5.myworkdayjobs.com/en-US/careers/job/United-States-
of-America---R emote/Software-Apps-Engineering-Intern_JR0021630?source=Linkedin', 5, 1);
```

```
INSERT INTO internship (Name, Application_Release, Application_Link, Location_ID,
Company_ID, Position_ID)
VALUES ('Digital Marketing Intern (Spring 2023)', '11/11/2022',
'https://careers.na.panasonic.com/careers/digital-marketing-intern-spring-2023?__jvsd=
LinkedIn&__jvst=Job%20Board&nl=1', 7, 6, 8);
```

```
INSERT INTO internship (Name, Application_Release, Hourly_Rate, Application_Link,
Location_ID, Company_ID, Position_ID)
VALUES ('Quantitative Researcher - Internship [2023 Summer]', '11/20/2022', 88,
'https://careers.twosigma.com/careers/JobDetail?jobId=10972&source=LinkedIn', 8, 7,
8);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('10/03/2022', 'In-Progress', 1, 1);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('12/03/2022', 'In-Progress', 1, 4);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('01/01/2023', 'Applied', 1, 5);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('12-12-2022', 'Applied', 2, 1);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'Accepted', 2, 2);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'Rejected', 3, 2);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'Accepted', 3, 3);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'Rejected', 4, 2);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-25-2022', 'Accepted', 4, 6);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'In-Progress', 5, 2);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'Rejected', 5, 3);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('12-13-2022', 'Accepted', 6, 7);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-13-2022', 'In-Progress', 7, 1);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-18-2022', 'Applied', 7, 2);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('12-14-2022', 'Accepted', 7, 5);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-29-2022', 'Rejected', 8, 4);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-14-2022', 'Accepted', 9, 1);
```

```
INSERT INTO student_application (Applied_Date, Status, Student_ID, Internship_ID)
VALUES ('11-23-2022', 'Applied', 10, 6);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (1, 3);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (1, 8);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (1, 10);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (2, 3);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (2, 10);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (3, 15);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (4, 7);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (4, 10);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (5, 10);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (6, 7);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (6, 12);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (7, 3);
```

```
INSERT INTO internship_requirement (Internship_ID, Requirement_ID)
VALUES (7, 9);
```

# Database Application

## Navigation Form

The screenshot shows a web application interface. On the left is a 'Star Main Menu' with a star icon and a list of links: Internship Form, Student Form, Location Form, Company Form, Position Form, Requirement Form, Student Application Report, Internship Report, and Internship Requirement Report. The 'Internship Form' link is highlighted. The main area displays the 'Internship Entry Form'. It contains several input fields: Internship ID (1), Internship Name (2023 Software Engineering Internship: Backend), Location ID (2), Company ID (1), Company Name (Netflix), Application Release (9/15/2022), Application Deadline (11/30/2022), Hourly Rate (USD) (60), Start Date (6/5/2023), End Date (8/24/2023), Position ID (1), Position Title (Software Engineer), Position Description (An IT professional who designs, develops, and maintain computer software at a company.), and Application Link (https://randomjobapplication.com/netflix/1). Below these fields is a table for 'Internship Requirement' with columns: Requirement\_ID, Category, and Condition. The table has three rows: 3 Grade Level Junior, 8 GPA 3.5+, and 10 Major Computer Science. At the bottom of the form are buttons for 'Add Record', 'Save Record', and 'Delete Record'.

The Navigation Form is the first form that appears when the database is opened. Different data entry forms and reports can be displayed by clicking on the selection on the left-hand side.

## Internship Entry Form

The screenshot shows the 'Internship Entry Form' in detail. It has the same layout as the previous screenshot, with input fields for Internship ID, Name, Location ID, Company ID, Name, Application Release, Deadline, Hourly Rate, Start/End Dates, Position ID, Title, Description, and Link. The 'Internship Requirement' table is also visible. The form includes 'Add Record', 'Save Record', and 'Delete Record' buttons at the bottom.

The Internship Entry Form is used to look up existing internships and input new internships into the database, alongside inputting new internship requirements as well. The Company Name, Position Title, and Position Description are already filled in depending on the Company\_ID and Position\_ID selected. The form also generates a new unique Internship\_ID when creating a new record. The form has several custom VBA codes to accomplish several functions as well.

```
Private Sub Internship_Name_AfterUpdate()  
    internship_Name = StrConv(internship_Name, vbProperCase)  
End Sub
```

This function is triggered when the Internship\_Name text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

```

Private Sub Application_Release_AfterUpdate()
    Dim releaseDate As Date
    Dim deadlineDate As Date
    If Not IsNull(Me.Application_Deadline) And Not IsNull(Me.Application_Release) Then
        releaseDate = DateValue(Me.Application_Release)
        deadlineDate = DateValue(Me.Application_Deadline)
        If deadlineDate <= releaseDate Then
            MsgBox "Error: The Application Release Date is later than the Application Deadline Date. Please try again.", vbCritical
            Me.Application_Release = Null
        End If
    End If
End Sub

```

This function is triggered when the Application\_Release text box is updated. It checks to see if the date set within the text box comes before the Application\_Deadline date chronologically. If it doesn't come before, the user will be prompted to enter a new Application\_Release date.

```

Private Sub Application_Deadline_AfterUpdate()
    Dim releaseDate As Date
    Dim deadlineDate As Date
    If Not IsNull(Me.Application_Release) And Not IsNull(Me.Application_Deadline) Then
        releaseDate = DateValue(Me.Application_Release)
        deadlineDate = DateValue(Me.Application_Deadline)
        If deadlineDate <= releaseDate Then
            MsgBox "Error: The Application Deadline Date is earlier than the Application Release Date. Please try again.", vbCritical
            Me.Application_Deadline = Null
        End If
    End If
End Sub

```

This function is triggered when the Application\_Deadline text box is updated. It checks to see if the date set within the text box comes after the Application\_Release date chronologically. If it doesn't come after, the user will be prompted to enter a new Application\_Deadline date.

```

Private Sub Start_Date_AfterUpdate()
    Dim startDate As Date
    Dim endDate As Date
    If Not IsNull(Me.End_Date) And Not IsNull(Me.Start_Date) Then
        startDate = DateValue(Me.Start_Date)
        endDate = DateValue(Me.End_Date)
        If endDate <= startDate Then
            MsgBox "Error: The Start Date is later than the End Date. Please try again.", vbCritical
            Me.Start_Date = Null
        End If
    End If
End Sub

```

This function is triggered when the Start\_Date text box is updated. It checks to see if the date set within the text box comes before the End\_Date chronologically. If it doesn't come before, the user will be prompted to enter a new Start\_Date.

```

Private Sub End_Date_AfterUpdate()
    Dim startDate As Date
    Dim endDate As Date
    If Not IsNull(Me.Start_Date) And Not IsNull(Me.End_Date) Then
        startDate = DateValue(Me.Start_Date)
        endDate = DateValue(Me.End_Date)
        If endDate <= startDate Then
            MsgBox "Error: The End Date is earlier than the Start Date. Please try again.", vbCritical
            Me.End_Date = Null
        End If
    End If
End Sub

```

This function is triggered when the End\_Date text box is updated. It checks to see if the date set within the text box comes after the Start\_Date chronologically. If it doesn't come after, the user will be prompted to enter a new End\_Date.

```

Private Sub Form_Current()
    Dim query As String
    ' Company Name
    If Not IsNull(Me.Company_ID.Value) Then
        query = "SELECT top 1 Name FROM Company WHERE Company_ID =" & Me.Company_ID.Value
        Set rs = CurrentDb.OpenRecordset(query)
        Me.company_Name.Value = rs!Name
    Else
        Me.company_Name.Value = ""
    End If
    ' Position Title and Description
    If Not IsNull(Me.Position_ID.Value) Then
        id = Me.Position_ID.Value
        query = "SELECT top 1 Position_Title FROM Position WHERE Position_ID =" & Me.Position_ID.Value
        Set rs = CurrentDb.OpenRecordset(query)
        Me.Position_Title.Value = rs!Position_Title
        query = "SELECT top 1 Description FROM Position WHERE Position_ID =" & Me.Position_ID.Value
        Set rs = CurrentDb.OpenRecordset(query)
        Me.Position_Description.Value = rs!Description
    Else
        Me.Position_Title.Value = ""
        Me.Position_Description.Value = ""
    End If
End Sub

```

This function is triggered when the focus moves to a record, making it the current record, or when the form is refreshed or requiered. It automatically fills in the Company Name, the Position Title, and the Position Description when selecting the Company\_ID and Position\_ID. If no ID is selected, then an empty string will be put in place, with respect to the Company\_ID and the Position\_ID.

```

Private Sub Company_ID_AfterUpdate()
    Dim query As String
    ' Company Name
    If Not IsNull(Me.Company_ID.Value) Then
        query = "SELECT top 1 Name FROM Company WHERE Company_ID =" & Me.Company_ID.Value
        Set rs = CurrentDb.OpenRecordset(query)
        Me.Company_Name.Value = rs!Name
    Else
        Me.Company_Name.Value = ""
    End If
End Sub

```

This function is triggered when a new selection is made for Company\_ID. It automatically fills in the Company Name associated with the selected Company\_ID.

```

Private Sub Position_ID_AfterUpdate()
    Dim query As String
    ' Position Title and Description
    If Not IsNull(Me.Position_ID.Value) Then
        id = Me.Position_ID.Value
        query = "SELECT top 1 Position_Title FROM Position WHERE Position_ID =" & Me.Position_ID.Value
        Set rs = CurrentDb.OpenRecordset(query)
        Me.Position_Title.Value = rs!Position_Title
        query = "SELECT top 1 Description FROM Position WHERE Position_ID =" & Me.Position_ID.Value
        Set rs = CurrentDb.OpenRecordset(query)
        Me.Position_Description.Value = rs!Description
    Else
        Me.Position_Title.Value = ""
        Me.Position_Description.Value = ""
    End If
End Sub

```

This function is triggered when a new selection is made for Position\_ID. It automatically fills in the Position Title and Description associated with the selected Position\_ID.



## Student Entry Form

The Student Entry Form is used to look up existing students and input new students into the database, alongside inputting new student applications as well. The form also generates a new unique Internship\_ID when creating a new record. The form has several custom VBA codes to accomplish several functions as well.

This function is triggered when the First\_Name text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

This function is triggered when the Last\_Name text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

This function is triggered when the Email text box is updated. It verifies to see if the inputted Email is valid in terms of the way it is formatted using regular expression. If the provided Email is not valid, the user will be prompted to enter a new one.

```
Private Sub School_AfterUpdate()  
    School = StrConv(School, vbProperCase)  
End Sub
```

This function is triggered when the School text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

```
Private Sub Major_AfterUpdate()  
    Major = StrConv(Major, vbProperCase)  
End Sub
```

This function is triggered when the Major text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

```
Private Sub Minor_AfterUpdate()  
    Minor = StrConv(Minor, vbProperCase)  
End Sub
```

This function is triggered when the Minor text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

## Location Entry Form



The Location Entry Form is used to query and add new internship locations to our database.

```
Private Sub Country_AfterUpdate()  
    Country = StrConv(Country, vbProperCase)  
End Sub
```

This function is triggered when the Country text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

```
Private Sub State_AfterUpdate()  
    State = StrConv(State, vbUpperCase)  
End Sub
```

This function is triggered when the State text box is updated. It converts the text within the text box to upper case.

```
Private Sub City_AfterUpdate()  
    City = StrConv(City, vbProperCase)  
End Sub
```

This function is triggered when the City text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

## Company Entry Form

A screenshot of a web form titled "Company Entry Form" in a dark blue header. The form contains five input fields: "Company ID" with the value "1", "Company Name" with "Netflix", "Employees" with "11300", "Headquarters" with "Los Gatos, California", and "Industry" with "Entertainment". At the bottom, there are three buttons: "Add Record", "Save Record", and "Delete Record".

Company ID	1
Company Name	Netflix
Employees	11300
Headquarters	Los Gatos, California
Industry	Entertainment

**Add Record** **Save Record** **Delete Record**

The Company Entry Form is used to query, update, and add new companies to our database.

```
Private Sub Company_Name_AfterUpdate()  
    company_Name = StrConv(company_Name, vbProperCase)  
End Sub
```

This function is triggered when the Company\_Name text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

```
Private Sub Headquarters_AfterUpdate()  
    Headquarters = StrConv(Headquarters, vbProperCase)  
End Sub
```

This function is triggered when the Headquarters text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

```
Private Sub Industry_AfterUpdate()  
    Industry = StrConv(Industry, vbProperCase)  
End Sub
```

This function is triggered when the Industry text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

## Position Entry Form

### Position Entry Form

Position ID	<input type="text" value="1"/>
Position Title	<input type="text" value="Software Engineer"/>
Description	<input type="text" value="An IT professional who designs, develops, and maintain computer software at a company."/>

Add RecordSave RecordDelete Record

The Position Entry Form is used to query, update, and add new internship positions to our database.

```
Private Sub Position_Title_AfterUpdate()  
    Position_Title = StrConv(Position_Title, vbProperCase)  
End Sub
```

This function is triggered when the Position\_Title text box is updated. It converts the text within the text box to proper case, which capitalizes the first letter and lower cases the rest of the letters of every word separated by spaces.

## Requirement Entry Form

### Requirement Entry Form

Requirement ID	<input type="text" value="1"/>
Category	<input type="text" value="Grade Level"/>
Condition	<input type="text" value="Freshman"/>

Add RecordSave RecordDelete Record

The Requirement Entry Form is used to query, update, and add requirements for internships to our database.

## Business Query: Most Popular Internships

Business Query: Most Popular Internships		
Internship_ID	Internship_Name	Number_Of_Applications
2	2023 Summer Financial Data Analyst Intern	5
1	2023 Software Engineering Internship: Backend	4
6	Digital Marketing Intern (Spring 2023)	2
5	Software Apps Engineerings Intern	2
4	Android Software Engineering Intern	2
3	2023 Technical Program Management Intern	2
7	Quantitative Researcher - Internship [2023 Summer]	1

**Business Question:** What are the most popular internships based on student applications?

This datasheet shows the most popular internships based on the number of student applications.

Based on this query:

```
SELECT internship.Internship_ID AS Internship_ID, internship.Name AS Internship_Name,  
       Count(student_application.Student_Application_ID) AS Number_Of_Applications  
FROM internship INNER JOIN student_application ON internship.Internship_ID = student_application.Internship_ID  
GROUP BY internship.Internship_ID, internship.Name  
ORDER BY Count(student_application.Student_Application_ID) DESC;
```

### Business Query: Most Popular Majors

Major	Number_Of_Students
Computer Information Systems	7
Statistics	1
Marketing Management	1
Computer Science	1

**Business Question:** What are the most popular majors based on students?

This datasheet shows the most popular majors based on the students on our platform.

Based on this query:

```
SELECT Major, Count(student.Major) AS Number_Of_Students  
FROM student  
GROUP BY student.Major  
ORDER BY Count(student.Major) DESC;
```

### Business Query: Most Popular Position

position_title	current_applicants
Software Engineer	8
Data Analyst	7
Marketing	2
Quantitative Analys	1

**Business Question:** What are the most popular positions based on student applications?

This datasheet shows the most popular positions based on student applications on our platform.

Based on this query:

```
SELECT position_title, COUNT(sa.internship_ID) AS current_applicants  
FROM (student_application AS sa INNER JOIN internship AS i ON sa.internship_ID = i.internship_ID)  
     INNER JOIN [position] AS p ON p.position_ID = i.position_ID  
GROUP BY position_title  
ORDER BY COUNT(sa.internship_ID) DESC;
```



## Business Query: Student Acceptance Rate

Business Query: Student Acceptance Rates			
Student_ID	First_Name	Last_Name	Acceptance_Rate
9	Jason	Schwartzman	1
6	Carlos	Silverio	1
4	Fawwad	Khan	0.5
3	Colleen	Zeng	0.5
2	Taohid	Shadat	0.5
7	Kyle	Chen	0.333333333333333
10	Riya	Sawant	0
8	Dhruv	Rajagopala	0
5	Jeffrey	Zheng	0
1	Efaz	Ahmed	0

**Business Question:** What are the acceptance rates per student based on student applications? This datasheet shows the rate at which students receive an internship based on the number of applications they've submitted on our platform.

Based on this query:

```
SELECT s.Student_ID, s.First_Name, s.Last_Name, (Count(If([t.Status]="Accepted",0))/Count(*)) AS Acceptance_Rate
FROM student AS s INNER JOIN student_application AS t ON s.Student_ID = t.Student_ID
GROUP BY s.Student_ID, s.First_Name, s.Last_Name
ORDER BY (Count(If([t.Status]="Accepted",0))/Count(*)) DESC;
```

## Student Application Report

Student Application Report						
ID	Student_Name	Status	Internship_ID	Internship_Name	Company	Position
1	Efaz Ahmed	In-Progress	4	Android Software Engineering Intern	Etsy	Software Engineer
		Applied	5	Software Apps Engineerings Intern	Yahoo	Software Engineer
		In-Progress	1	2023 Software Engineering Internship: Backend	Netflix	Software Engineer
2	Taohid Shadat	Applied	1	2023 Software Engineering Internship: Backend	Netflix	Software Engineer
		Accepted	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
		Rejected	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
3	Colleen Zeng	Accepted	3	2023 Technical Program Management Intern	YouTube	Data Analyst
		Rejected	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
		Accepted	3	2023 Technical Program Management Intern	YouTube	Data Analyst
4	Fawwad Khan	Accepted	6	Digital Marketing Intern (Spring 2023)	Panasonic	Marketing
		Rejected	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
		Rejected	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
5	Jeffrey Zheng	Rejected	3	2023 Technical Program Management Intern	YouTube	Data Analyst
		In-Progress	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
		Accepted	7	Quantitative Researcher - Internship [2023 Summer]	Two Sigma	Quantitative Analyst
6	Carlos Silverio	In-Progress	1	2023 Software Engineering Internship: Backend	Netflix	Software Engineer
		Applied	2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst
		Accepted	5	Software Apps Engineerings Intern	Yahoo	Software Engineer

This report shows each student and the internships they have applied to. It also shows the application status of each application, and the company and position of every internship opportunity.

Based on this query:

```
SELECT s.Student_ID AS ID, s.First_Name & " " & s.Last_Name AS Student_Name,
       sa.Status AS Status, i.Internship_ID AS Internship_ID, i.Name AS Internship_Name,
       c.Name AS Company, p.Position_Title AS [Position]
FROM
    (((student AS s INNER JOIN student_application AS sa ON s.Student_ID = sa.Student_ID)
     INNER JOIN internship AS i ON sa.Internship_ID = i.Internship_ID)
     INNER JOIN [position] AS p ON i.Position_ID = p.Position_ID)
     INNER JOIN company AS c ON i.Company_ID = c.Company_ID;
```

## Internship Report

Wednesday, December 14, 2022						
8:08:22 PM						
Internship_ID	Internship_Name	Company_Name	Position_Title	Country	State	City
1	2023 Software Engineering Internship: Backend	Netflix	Software Engineer	United States	NY	New York
2	2023 Summer Financial Data Analyst Intern	YouTube	Data Analyst	United States	NY	New York
3	2023 Technical Program Management Intern	YouTube	Data Analyst	United States	NY	New York
4	Android Software Engineering Intern	Etsy	Software Engineer	United States	NY	Brooklyn
5	Software Apps Engineerings Intern	Yahoo	Software Engineer			
6	Digital Marketing Intern (Spring 2023)	Panasonic	Marketing	United States	NJ	Newark
7	Quantitative Researcher - Internship [2023 Summer]	Two Sigma	Quantitative Analyst	United States	NY	New York

Page 1 of 1

This report shows every internship opportunity with its associated company name, position title, and the country, state, and city of its location.

Based on this query:

```
SELECT i.Internship_ID AS Internship_ID, i.Name AS Internship_Name,
       c.Name AS Company_Name, p.Position_Title AS Position_Title, l.Country, l.State, l.City
FROM
    (
        (Internship AS i INNER JOIN Company AS c ON i.Company_ID = c.Company_ID)
        LEFT JOIN Location AS l ON i.Location_ID = l.Location_ID)
     INNER JOIN [Position] AS p ON i.Position_ID = p.Position_ID;
```

## Internship Requirement Report

Internship Requirement Report				
Internship_ID Name		Requirement_ID Category		Condition
1	2023 Software Engineering Internship: Backend	10	Major	Computer Science
		8	GPA	3.5+
		3	Grade Level	Junior
2	2023 Summer Financial Data Analyst Intern	10	Major	Computer Science
		3	Grade Level	Junior
3	2023 Technical Program Management Intern	15	Major	Management
4	Android Software Engineering Intern	10	Major	Computer Science
		7	GPA	3.0+
5	Software Apps Engineerings Intern	10	Major	Computer Science
6	Digital Marketing Intern (Spring 2023)	12	Major	Marketing
		7	GPA	3.0+
7	Quantitative Researcher - Internship [2023 Summer]	9	GPA	4.0
		3	Grade Level	Junior
Wednesday, December 14, 2022				
Page 1 of 1				

This report shows every internship and all of the requirements associated with each internship.

Based on this query:

```
SELECT i.Internship_ID, i.Name, r.Requirement_ID, r.Category, r.Condition
FROM
  (internship AS i INNER JOIN internship_requirement AS ir ON i.Internship_ID = ir.Internship_ID)
INNER JOIN requirement AS r ON ir.Requirement_ID = r.Requirement_ID;
```



## **Conclusion**

Overall, the project was an interesting challenge I took on. The steps within the project that were the easiest would be creating the conceptual and logical models. I was provided with many materials and practice to take on this task at ease, and with minimal complications. However, the journey from creating the E-R model to optimizing it, to writing the SQL code, and lastly to building the application was difficult with many obstacles to overcome. Some of those obstacles include debugging VBA code in Microsoft Access, being in compliance with Microsoft Access data types, creating Master/Detail forms, and handling intersection entities.

Despite facing many challenging obstacles, I learned a lot from this experience. I learned how to navigate through Microsoft Access by learning how to create different types of forms, such as Master/Detail forms, learning how to code in the VBA programming language, and how to create Access-specific SQL queries. I also learned how to properly implement the steps in converting a conceptual model to a physical model.

One thing I would change about my project is reducing the number of abstractions. For example, due to creating a Location entity, I am now required to input a location for the internship before inputting the internship itself. This would result in creating more steps to complete the same process and would lead to an unmaintainable DBMS. Implementing changes to create fewer abstractions would not only lead to a more maintainable DBMS, but it can create fewer opportunities for error in the future. It would also impact the application by reducing the number of tables in the DBMS.

Another thing I would change about my project is by adding a “status” field to the internship table. This field would represent the internship’s availability in terms of whether it's “open” or “closed.” If the internship is “open” then the student would be allowed to apply to the said internship. However, if it's closed, then the student should not be allowed to apply. These changes can be implemented through VBA code by observing the current date and the date of the Application Deadline. When the current date is greater than the Application Deadline, the status of the internship will become “closed.” This will impact the application by incorporating more business logic into the application itself, and by reducing future errors with students applying to internships that are no longer available.

This project provided me with vast insight into how databases work in the real world. When I started the project, I planned it to be within the scope of a startup, however, after learning so much more about DBMS and databases, I hope to turn the Star Internship Finder into a full-fledged corporation. Finding internships can be very difficult, especially as a student working in a competitive industry. I hope that my platform will guide those to finding their dream jobs.