CAREER LAUNCHPAD

TEAM MEMBERS:
1-YASSIN TAMER
2-EZZ ELDIN MOHAMED
3-YASSIN MOHAMED

Deliverable #1 Template – Database Project (Fall 2025)

1. English Requirements (Business Rules)

The **Career Launchpad System** is a unified digital platform designed to connect **students**, **companies offering internships**, **recruiters**, and **university career services**.
The platform ensures that all internship opportunities visible to students are **reviewed and approved by the university**, maintaining academic quality, legal compliance, and relevance.

> Write your own system requirements here:

(1)Over view on business scenario

The Career Launchpad system is a unified platform that connects students, company that afford internship exactly with recruiter, and university career services (it's the rule that accept internship to be appear on university profile).

Students use the platform to search and apply for internships.

Recruiters post internship opportunities and manage applicants.

The university reviews and approves postings to maintain quality and compliance.

Executives (like the Dean) can access high-level reports about student placement trends.

(2)Functional requirements and Business rules (constraints)

R1 — User Registration & Rules

R1.1 The system shall allow students, company recruiters, and university staff to create accounts.

R1.2 Each user must have exactly one role (Student / Recruiter / Staff / Executive).

R1.3 A user cannot access features outside their role.

R2 — Student Profile Rules

R2.1 A student may upload multiple supporting documents, but must mark one CV as primary.

R2.2 Students may update their profile at any time.

R2.3 A student may not apply for an internship if their profile is incomplete.

R3 — Company & Recruiter Rules

R3.1 Each recruiter must belong to one company.

R3.2 Every post created by recruiter must be reviewed by the university before students can see it.

R3.3 Recruiters can accept or reject student applicants from the short list.

R4 — Internship Posting Requirements

R4.1 An internship posting must include:

Title – Description - Required skills - Internship duration - Application deadline -Company information

R4.2 A posting cannot be published unless approved by the university staff

R5 — University Approval Workflow

R5.1 Career Services staff must review every new posting.

R5.2 Staff may approve, reject, or request modifications.

R5.3 A rejected posting must include a reason/comment.

R5.4 Only approved postings become visible to students.

R5.5 Staff cannot edit recruiter-created postings; they can only approve or reject

R6 — Internship Search & Application

R6.1 Students may search for internships using filters (location, skills, duration, etc.).

R6.2 A student apply to an internship only once.

R6.3 Applications attach the student's current profile and CV.

R6.4 The system must confirm successful application submission.

R6.5 Students may track application status: Submitted -> In Review -> Shortlisted -> Accepted -> Rejected

R7 — Application Management (Recruiter Side)

R7.1 Recruiters may view all applicants for their postings.

R7.2 Recruiters may view each applicant's full student profile.

R7.3 Recruiters may change application status.

R7.4 When status changes: The system must notify the student automatically.

R8 — Messaging System

R8.1 Students and recruiters may exchange messages only if: A student has applied to that recruiter's

internship.

R8.2 The system must notify users when a new message arrives.

R8.3 Message history must be saved and viewable.

R9 — Executive Reports (Head of Career Services / Dean)

R9.1 Executives may generate high-level reports including:

Total placements - Active internships - Top partner companies - Industry distribution of postings

R9.2 Reports may be downloaded as PDF or CSV.

R9.3 Dean make reports about the students rejected or students in the internship right now or the most

effective internship companies and a lot of other things that need to be analyzed

R10 — System Automation & Constraints

R10.1 When an internship reaches its application deadline, the system must automatically set the status to

Closed.

R10.2 The system must log all important actions (post creation, approval, status changes).

R10.3 Notifications must be sent for:

Application status changes

New postings approved

Messages received

R11 — Constraints & Validation Rules

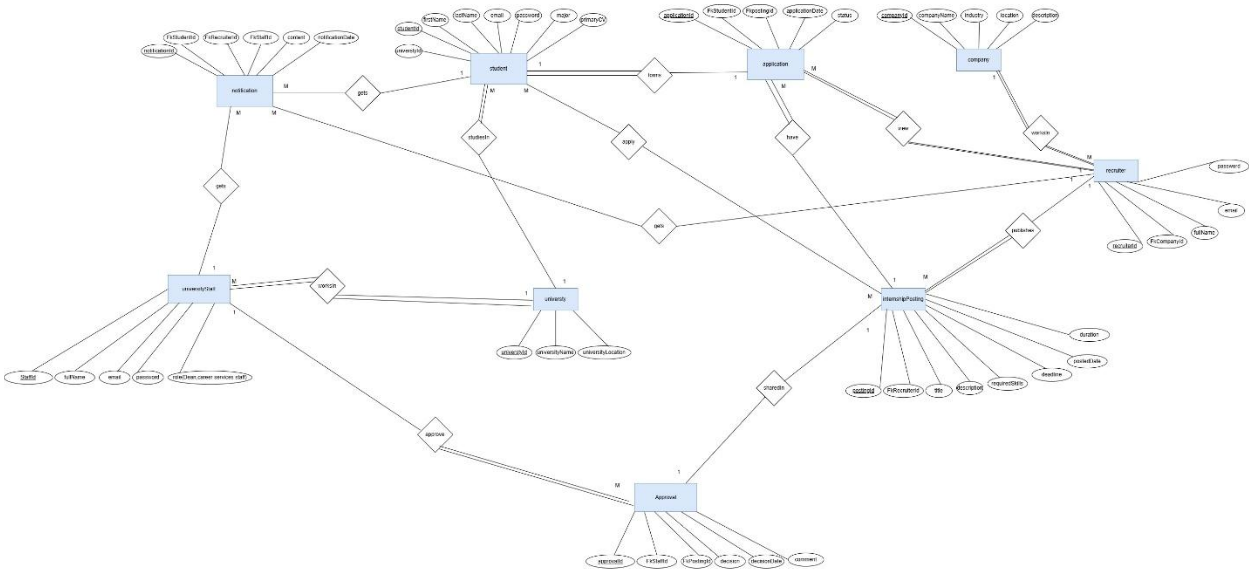
R11.1 A posting must have a deadline that is after the posting date.

R11.2 Eeach company must have one recruiter to post internships.

R11.3 Students cannot modify an application after submitting it.

R11.4 Each account if it for student or recruiter or university each account has its access rules

2. Entity–Relationship Diagram (ERD)



3. Relational Schema

For each entity and relationship, convert the ERD into a relational schema.

Your schema must include: - Primary keys (PK) - Foreign keys (FK) - Composite/multivalued attributes (if any), mapped correctly

Write your own schema here:

notification (PKnotificationId, content, notification date, FKstudentId, FKstaffId, FKrecruiterId)

student (PKstudentId, firstName, lastName, email, password, major, primaryCV,FK universityId, FKapplicationId)

application (PKapplicationId, application date, status, FKpostingId,FKrecruiterId)

company (PKcompanyId, company name, industry, location, description)

recruiter (PKrecruiterId, fullName, email, password,FKcompanyId)

internship posting (PKpostingId, title, description, required skills, deadline, posted date, duration, FKrecruiterId) approval (PKapprovalId, decision, decision date, comment, FKpostingId, FKstaffId)

university staff (PKstaffId, fullName, email, password, role, FKuniversityId)

university (PKuniversityId, university name, university location)

apply (PKstudentId, PKpostingId)

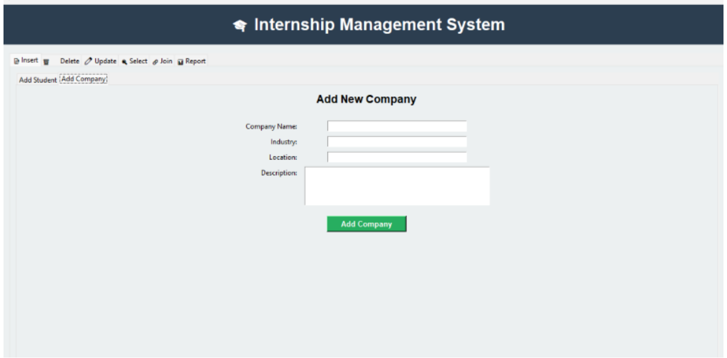
4. list of all queries (not the ad hoc ones)

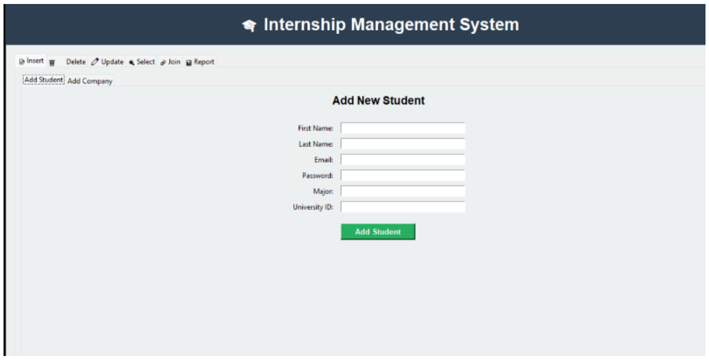
```
-- Query 1: Application statistics by company
--Count Applications per Company with Average Status
--Purpose: Show how many applications each company received and calculate statistics
about application status.
SELECT
    c.companyId,
    c.company_name,
    c.industry,
    COUNT(a.applicationId) AS total_applications,
    COUNT(CASE WHEN a.status = 'Accepted' THEN 1 END) AS accepted_count,
    COUNT(CASE WHEN a.status = 'Pending' THEN 1 END) AS pending_count,
    COUNT(CASE WHEN a.status = 'Rejected' THEN 1 END) AS rejected_count
FROM company c
JOIN recruiter r ON c.companyId = r.companyId
JOIN internship_posting ip ON r.recruiterId = ip.recruiterId
JOIN application a ON ip.postingId = a.postingId
GROUP BY c.companyId, c.company_name, c.industry
HAVING COUNT(a.applicationId) > 0
ORDER BY total_applications DESC;

-- Query 2: Average applications per student grouped by major
--Average Applications per Student by Major
--Purpose: Analyze which majors have students applying to more internships on
average.
SELECT
    s.major,
    COUNT(DISTINCT s.studentId) AS total_students,
    COUNT(a.applicationId) AS total_applications,
    CAST(COUNT(a.applicationId) AS FLOAT) / COUNT(DISTINCT s.studentId) AS
avg_applications_per_student,
    MAX(application_date) AS most_recent_application
FROM student s
LEFT JOIN application a ON s.studentId = a.studentId
GROUP BY s.major
HAVING COUNT(a.applicationId) > 0
ORDER BY avg_applications_per_student DESC;

-- Query 3: Students applying more than average (subquery in WHERE)
--Students Who Applied to More Postings Than Average
--Purpose: Find high-activity students who are applying to more internships than
their peers.
SELECT
    s.studentId,
    s.firstName,
```

```sql
    s.lastName,
    s.email,
    s.major,
    COUNT(app.applicationId) AS application_count
FROM student s
JOIN application app ON s.studentId = app.studentId
GROUP BY s.studentId, s.firstName, s.lastName, s.email, s.major
HAVING COUNT(app.applicationId) > (
    SELECT AVG(app_count)
    FROM (
        SELECT COUNT(applicationId) AS app_count
        FROM application
        GROUP BY studentId
    ) AS subquery
)
ORDER BY application_count DESC;

-- Query 4: Postings without applications (subquery with NOT EXISTS)
--nternship Postings With No Applications
--Purpose: Find postings that haven't received any applications yet (may need
promotion or review).
SELECT
    ip.postingId,
    ip.title,
    ip.description,
    ip.required_skills,
    ip.deadline,
    ip.posted_date,
    r.fullName AS recruiter_name,
    c.company_name
FROM internship_posting ip
JOIN recruiter r ON ip.recruiterId = r.recruiterId
JOIN company c ON r.companyId = c.companyId
WHERE NOT EXISTS (
    SELECT 1
    FROM application a
    WHERE a.postingId = ip.postingId
)
AND ip.deadline >= GETDATE()
ORDER BY ip.deadline ASC;


-- Query 5: Comprehensive application view joining 5 tables
--Complete Application Details (5 Tables)
--Purpose: Show full application information including student, posting, company,
and approval details.
SELECT
    a.applicationId,
    a.application_date,
    a.status,
    s.firstName + ' ' + s.lastName AS student_name,
    s.email AS student_email,
    s.major,
    ip.title AS internship_title,
    ip.required_skills,
    c.company_name,
    c.industry,
    c.location AS company_location,
    r.fullName AS recruiter_name,
    ap.decision AS approval_decision,
    ap.decision_date
FROM application a
JOIN student s ON a.studentId = s.studentId
JOIN internship_posting ip ON a.postingId = ip.postingId
JOIN recruiter r ON ip.recruiterId = r.recruiterId
JOIN company c ON r.companyId = c.companyId
LEFT JOIN approval ap ON ip.postingId = ap.postingId
ORDER BY a.application_date DESC;

-- Query 6: Staff approval activity with posting and company details (4 tables)
--University Staff Monitoring (4 Tables)
--Purpose: Show which staff members approved which postings and track their decision
patterns.
SELECT
    us.staffId,
    us.fullName AS staff_name,
    us.role,
    u.university_name,
    COUNT(ap.approvalId) AS total_approvals,
    SUM(CASE WHEN ap.decision = 'Approved' THEN 1 ELSE 0 END) AS approved_count,
    SUM(CASE WHEN ap.decision = 'Rejected' THEN 1 ELSE 0 END) AS rejected_count,
    ip.title AS recent_posting_reviewed,
    c.company_name AS recent_company
FROM university_staff us
JOIN university u ON us.universityId = u.universityId
LEFT JOIN approval ap ON us.staffId = ap.staffId
LEFT JOIN internship_posting ip ON ap.postingId = ip.postingId
LEFT JOIN recruiter r ON ip.recruiterId = r.recruiterId
LEFT JOIN company c ON r.companyId = c.companyId
GROUP BY us.staffId, us.fullName, us.role, u.university_name, ip.title,
c.company_name
ORDER BY total_approvals DESC;

-- Query 7: Students who haven't applied to any internships
--Students Without Applications (Join + Filter)
--Purpose: Identify students who haven't applied to any internships (may need
guidance).
SELECT
    s.studentId,
    s.firstName,
    s.lastName,
    s.email,
    s.major,
    u.university_name
FROM student s
JOIN university u ON s.universityId = u.universityId
LEFT JOIN application a ON s.studentId = a.studentId
WHERE a.applicationId IS NULL
ORDER BY s.major, s.lastName;

-- Query 8: Recruiter performance with posting and application metrics
--Purpose: Analyze recruiter effectiveness - postings created, applications
received, and approval rates.
SELECT
    r.recruiterId,
    r.fullName AS recruiter_name,
  r.email,
  c.company_name,
  COUNT(DISTINCT ip.postingId) AS total_postings,
  COUNT(DISTINCT a.applicationId) AS total_applications_received,
  CAST(COUNT(DISTINCT a.applicationId) AS FLOAT) /
    NULLIF(COUNT(DISTINCT ip.postingId), 0) AS avg_applications_per_posting,
  COUNT(DISTINCT ap.approvalId) AS postings_reviewed,
  SUM(CASE WHEN ap.decision = 'Approved' THEN 1 ELSE 0 END) AS approved_postings
FROM recruiter r
JOIN company c ON r.companyId = c.companyId
LEFT JOIN internship_posting ip ON r.recruiterId = ip.recruiterId
LEFT JOIN application a ON ip.postingId = a.postingId
LEFT JOIN approval ap ON ip.postingId = ap.postingId
GROUP BY r.recruiterId, r.fullName, r.email, c.company_name
HAVING COUNT(DISTINCT ip.postingId) > 0
ORDER BY total_applications_received DESC;
```

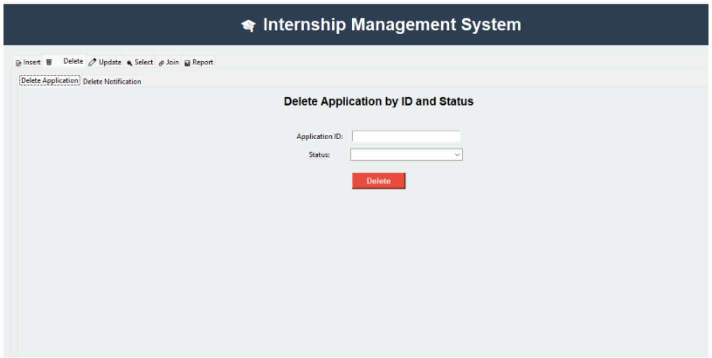
5. The GUI sceenshots that performs at least:

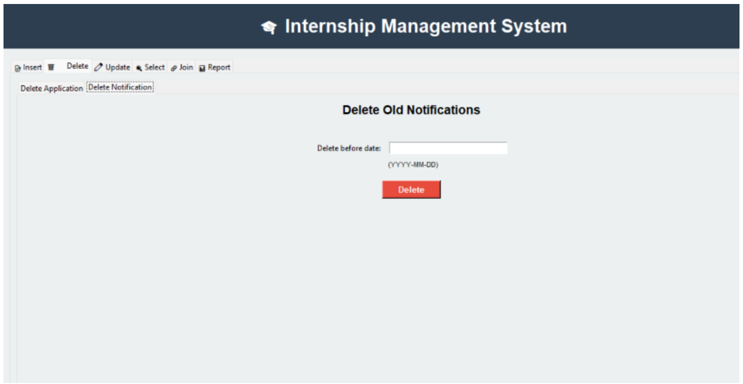
- 2 Insert Statements on 2 different tables.



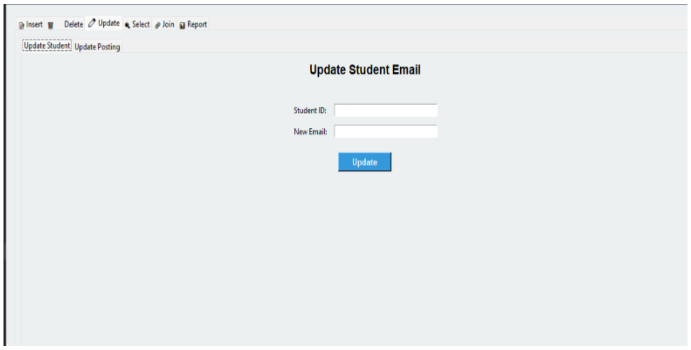


- 2 Delete Statements on 2 different tables (with conditions).





- 2 Update Statements on 2 different tables [with condition].

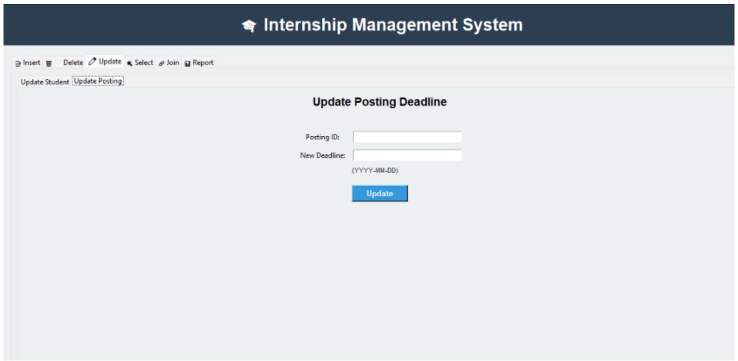


Update Student Email

Student ID:

New Email:

Update



Update Posting Deadline

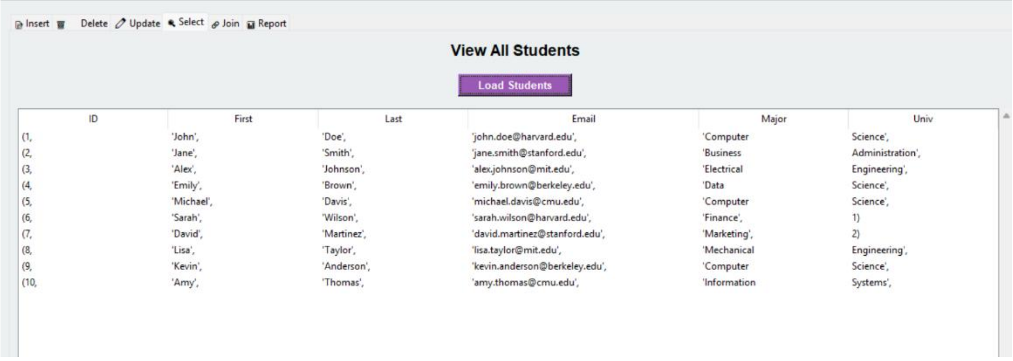
Posting ID:

New Deadline:

(YYYY-MM-DD)

Update

- Select data from any table(s) of the database.

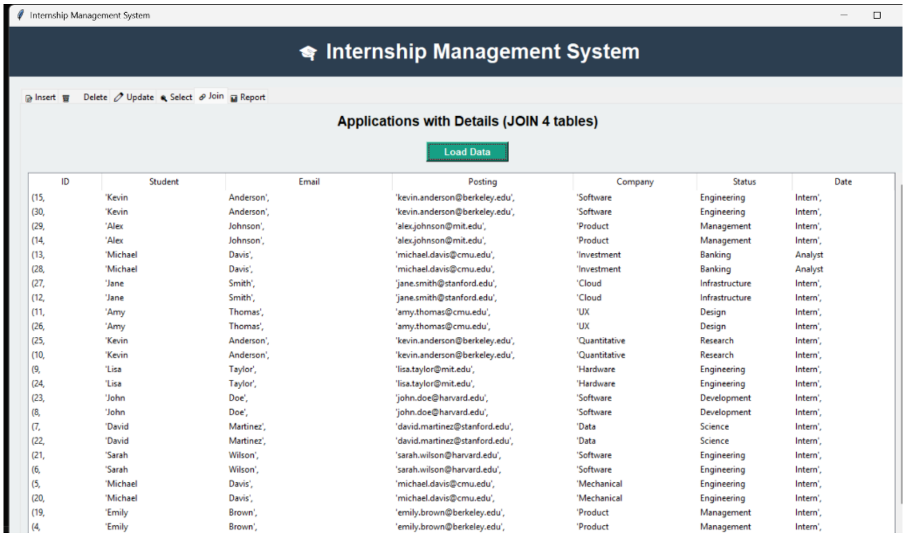


View All Students

Load Students

ID	First	Last	Email	Major	Univ
(1,	'John',	'Doe',	'john.doe@harvard.edu',	'Computer	Science',
(2,	'Jane',	'Smith',	'jane.smith@stanford.edu',	'Business	Administration',
(3,	'Alex',	'Johnson',	'alex.johnson@mit.edu',	'Electrical	Engineering',
(4,	'Emily',	'Brown',	'emily.brown@berkeley.edu',	'Data	Science',
(5,	'Michael',	'Davis',	'michael.davis@cmu.edu',	'Computer	Science',
(6,	'Sarah',	'Wilson',	'sarah.wilson@harvard.edu',	'Finance',	1)
(7,	'David',	'Martinez',	'david.martinez@stanford.edu',	'Marketing',	2)
(8,	'Lisa',	'Taylor',	'lisa.taylor@mit.edu',	'Mechanical	Engineering',
(9,	'Kevin',	'Anderson',	'kevin.anderson@berkeley.edu',	'Computer	Science',
(10,	'Amy',	'Thomas',	'amy.thomas@cmu.edu',	'Information	Systems',

- Select data that involves more than one table of the database- (using joins).

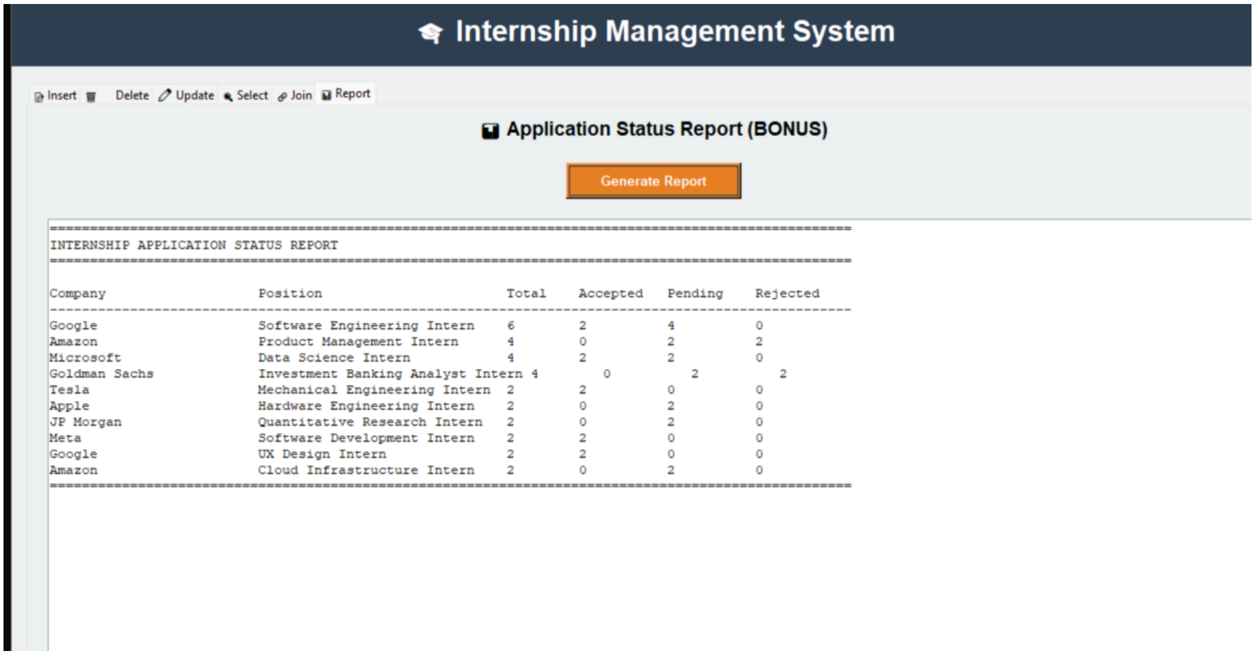


Applications with Details (JOIN 4 tables)

Load Data

ID	Student	Email	Posting	Company	Status	Date
(15,	'Kevin	Anderson',	'kevin.anderson@berkeley.edu',	'Software	Engineering	Intern',
(30,	'Kevin	Anderson',	'kevin.anderson@berkeley.edu',	'Software	Engineering	Intern',
(29,	'Alex	Johnson',	'alex.johnson@mit.edu',	'Product	Management	Intern',
(14,	'Alex	Johnson',	'alex.johnson@mit.edu',	'Product	Management	Intern',
(13,	'Michael	Davis',	'michael.davis@cmu.edu',	'Investment	Banking	Analyst
(28,	'Michael	Davis',	'michael.davis@cmu.edu',	'Investment	Banking	Analyst
(27,	'Jane	Smith',	'jane.smith@stanford.edu',	'Cloud	Infrastructure	Intern',
(12,	'Jane	Smith',	'jane.smith@stanford.edu',	'Cloud	Infrastructure	Intern',
(11,	'Amy	Thomas',	'amy.thomas@cmu.edu',	'UX	Design	Intern',
(26,	'Amy	Thomas',	'amy.thomas@cmu.edu',	'UX	Design	Intern',
(25,	'Kevin	Anderson',	'kevin.anderson@berkeley.edu',	'Quantitative	Research	Intern',
(10,	'Kevin	Anderson',	'kevin.anderson@berkeley.edu',	'Quantitative	Research	Intern',
(9,	'Lisa	Taylor',	'lisa.taylor@mit.edu',	'Hardware	Engineering	Intern',
(24,	'Lisa	Taylor',	'lisa.taylor@mit.edu',	'Hardware	Engineering	Intern',
(23,	'John	Doe',	'john.doe@harvard.edu',	'Software	Development	Intern',
(8,	'John	Doe',	'john.doe@harvard.edu',	'Software	Development	Intern',
(7,	'David	Martinez',	'david.martinez@stanford.edu',	'Data	Science	Intern',
(22,	'David	Martinez',	'david.martinez@stanford.edu',	'Data	Science	Intern',
(21,	'Sarah	Wilson',	'sarah.wilson@harvard.edu',	'Software	Engineering	Intern',
(6,	'Sarah	Wilson',	'sarah.wilson@harvard.edu',	'Software	Engineering	Intern',
(5,	'Michael	Davis',	'michael.davis@cmu.edu',	'Mechanical	Engineering	Intern',
(20,	'Michael	Davis',	'michael.davis@cmu.edu',	'Mechanical	Engineering	Intern',
(19,	'Emily	Brown',	'emily.brown@berkeley.edu',	'Product	Management	Intern',
(4,	'Emily	Brown',	'emily.brown@berkeley.edu',	'Product	Management	Intern',

- Generate 1 meaningful report. Bonus


Internship Management System
Insert
Delete
Update
Select
Join
Report
Application Status Report (BONUS)
Generate Report

```
====================================================================================================
INTERNSHIP APPLICATION STATUS REPORT
====================================================================================================

Company                   Position                       Total    Accepted   Pending    Rejected
----------------------------------------------------------------------------------------------------
Google                    Software Engineering Intern    6        2          4          0
Amazon                    Product Management Intern      4        0          2          2
Microsoft                 Data Science Intern            4        2          2          0
Goldman Sachs             Investment Banking Analyst Intern 4         0          2          2
Tesla                     Mechanical Engineering Intern  2        2          0          0
Apple                     Hardware Engineering Intern    2        0          2          0
JP Morgan                 Quantitative Research Intern   2        0          2          0
Meta                      Software Development Intern    2        2          0          0
Google                    UX Design Intern               2        2          0          0
Amazon                    Cloud Infrastructure Intern    2        0          2          0
====================================================================================================
```