

PROJECT 1

-By Xiaodi Wand, Udhaya Bhuvanesh

1. Problem Statement

College students applying for internships often manage dozens of applications across multiple companies and job platforms simultaneously. Without a structured system, it becomes easy to miss deadlines, forget interview times, or lose track of where each application stands. This database tracks the full lifecycle of an internship application from the moment a student submits their materials to a final hiring decision.

2. Business Rules

- A student has a profile with their name, university, major, and graduation year.
- A student can submit many applications, but each application belongs to exactly one student.
- Each application targets exactly one job posting at a company.
- A company can have many job postings, but each posting belongs to exactly one company.
- Each job posting has a title, location (remote/onsite/hybrid), and job type (internship/co-op/part-time).
- An application has a submission date, current status (Applied, Online Assessment(OA), Interviewing, Offer, Rejected, Withdrawn), and optional notes.
- A student may go through multiple interview rounds for the same application (e.g., phone screen, technical, behavioral, final).
- Each interview round has a scheduled date, interview type, and outcome (Pending, Passed, Failed).
- A student can store contacts at a company (e.g., recruiters, hiring managers) and associate them with an application.
- A student can tag an application with one or more tags (e.g., "Dream Company", "Referral", "High Priority") for personal organization.
- An application can have at most one offer record, which stores compensation details and an offer deadline.

3. Nouns

- Student
- Company
- JobPosting
- Application
- InterviewRound
- Tag

- ApplicationTag
- Contact
- ApplicationContact
- Offer
- JobBoard
- StatusHistory

4. Actions

- Submits
- Targets
- Schedules
- Tags

5. Relational Schema and Normalization (BCNF) (Q3)

Relational Schema:

- Student
Student(**student_id**, name, university, major, graduation_year)
- Company
Company(**company_id**, company_name, industry, hq_location)
- JobPosting
JobPosting(**job_id**, company_id → Company.company_id, title, location_mode, job_type)
- Application
Application(**app_id**, student_id → Student.student_id, job_id → JobPosting.job_id, submission_date, current_status, notes)
- InterviewRound
InterviewRound(**round_id**, app_id → Application.app_id, round_type, scheduled_date, outcome, notes)
- Tag
Tag(**tag_id**, tag_name)
- ApplicationTag
ApplicationTag(
 app_id → Application.app_id,
 tag_id → Tag.tag_id,
 PRIMARY KEY (app_id, tag_id)
)
- Contact
Contact(**contact_id**, company_id → Company.company_id, contact_name, role, email, phone)

- ApplicationContact
ApplicationContact(
 app_id → Application.app_id,
 contact_id → Contact.contact_id,
 PRIMARY KEY (app_id, contact_id)
)
- Offer
Offer(
 offer_id,
 app_id UNIQUE → Application.app_id,
 compensation,
 deadline,
 offer_notes
)
- JobBoard
JobBoard(**board_id**, board_name, url)
- StatusHistory
StatusHistory(
 history_id,
 app_id → Application.app_id,
 status,
 changed_at,
 notes
)

This schema reflects the following relationships described in the business rules:

- A student can submit many applications, but each application belongs to exactly one student.
- A company can have many job postings.
- Each application targets exactly one job posting.
- An application can have multiple interview rounds.
- An application can have one or more tags.
- A student can store contacts at a company and associate them with an application.
- An application can have at most one offer record.

Functional Dependencies and BCNF Proof:

Student

FDs:

student_id → name, university, major, graduation_year

Candidate Key:

{student_id}

Since the only determinant is a candidate key, Student is in BCNF.

Company

FDs:

company_id \rightarrow company_name, industry, hq_location

Candidate Key:

{company_id}

All determinants are superkeys.

Therefore, Company is in BCNF.

JobPosting

FDs:

job_id \rightarrow company_id, title, location_mode, job_type

Candidate Key:

{job_id}

All determinants are superkeys.

Therefore, JobPosting is in BCNF.

Application

FDs:

app_id \rightarrow student_id, job_id, submission_date, current_status, notes

Candidate Key:

{app_id}

All determinants are superkeys.

Therefore, Application is in BCNF.

InterviewRound

FDs:

round_id \rightarrow app_id, round_type, scheduled_date, outcome, notes

Candidate Key:
{round_id}

All determinants are superkeys.
Therefore, InterviewRound is in BCNF.

Tag

FDs:
 $\text{tag_id} \rightarrow \text{tag_name}$

Candidate Key:
{tag_id}

All determinants are superkeys.
Therefore, Tag is in BCNF.

ApplicationTag

FDs:
 $(\text{app_id}, \text{tag_id}) \rightarrow \emptyset$

Candidate Key:
{app_id, tag_id}

There are no non-trivial dependencies other than the composite key.
Therefore, ApplicationTag is in BCNF.

Contact

FDs:
 $\text{contact_id} \rightarrow \text{company_id}, \text{contact_name}, \text{role}, \text{email}, \text{phone}$

Candidate Key:
{contact_id}

All determinants are superkeys.
Therefore, Contact is in BCNF.

ApplicationContact

FDs:
 $(\text{app_id}, \text{contact_id}) \rightarrow \emptyset$

Candidate Key:

{app_id, contact_id}

There are no non-trivial dependencies other than the composite key.

Therefore, ApplicationContact is in BCNF.

Offer

FDs:

offer_id \rightarrow app_id, compensation, deadline, offer_notes

app_id \rightarrow offer_id, compensation, deadline, offer_notes

Candidate Keys:

{offer_id}

{app_id}

Because app_id is declared UNIQUE, both offer_id and app_id function as candidate keys. Therefore, all determinants are superkeys.

Therefore, Offer is in BCNF.

JobBoard

FDs:

board_id \rightarrow board_name, url

Candidate Key:

{board_id}

Since the only determinant is a candidate key, JobBoard is in BCNF.

StatusHistory

FDs:

history_id \rightarrow app_id, status, changed_at, notes

Candidate Key:

{history_id}

All determinants are superkeys.

Therefore, StatusHistory is in BCNF.