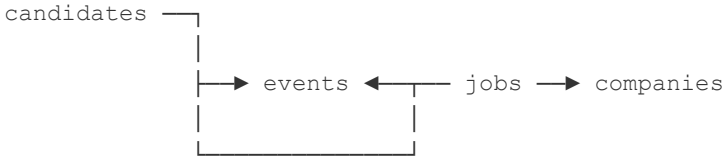
JobMatch Data Dictionary

BADM 576 — Spring 2026

Overview

JobMatch is a recruiting platform simulation. The dataset contains four tables representing candidates (job seekers), companies (employers), jobs (postings), and events (all platform activity).

Table Relationships



- `events.candidate_id` → `candidates.candidate_id`
- `events.job_id` → `jobs.job_id`
- `jobs.company_id` → `companies.company_id`

Table 1: candidates.csv

Rows: 5,000 • **Unit of observation:** One job seeker registered on the platform

Column	Type	Description
candidate_id	string	Primary key (format: CAND-XXXX)
name	string	Candidate name (synthetic)
experience_years	integer	Years of work experience (0–20)
education_level	string	Highest degree: High School, Associate, Bachelor, Master, PhD
in_salary_expectation	integer	Candidate's minimum acceptable salary (hidden from employers)
in_preferred_seniority	string	Preferred job level: Entry, Mid, Senior, Lead
in_preferred_industries	string	Comma-separated list of preferred industries
in_preferred_locations	string	Comma-separated list of preferred cities
in_true_skills	string	Candidate's actual skills (hidden — platform only sees resume)
resume_skills_raw	string	Skills listed on resume (may differ from true skills)
in_quality_score	float	Platform's internal quality score (0–1, hidden from all parties)

Note: Columns prefixed with `in_` are **internal/hidden** — they represent ground truth that would not be visible to the other party in a real platform.

Table 2: companies.csv

Rows: 200 • **Unit of observation:** One employer registered on the platform

Column	Type	Description
company_id	string	Primary key (format: COMP-XXX)
name	string	Company name (synthetic)
industry	string	Primary industry
size	string	Company size: Startup, SMB, Mid-Market, Enterprise

Industry values: Consulting, Education, Energy, Finance, Healthcare, Manufacturing, Media, Retail, Technology, Transportation

Size values: Startup, SMB, Mid-Market, Enterprise

Table 3: jobs.csv

Rows: 1,500 • **Unit of observation:** One job posting on the platform

Column	Type	Description
job_id	string	Primary key (format: JOB-XXXX)
company_id	string	Foreign key → companies.company_id
title	string	Job title
seniority	string	Required level: Entry, Mid, Senior, Lead
industry	string	Job's industry (matches company industry)
required_skills	string	Comma-separated list of required skills
location	string	Job location (city)
in_salary_min	integer	Minimum salary budget (hidden from candidates)
in_salary_max	integer	Maximum salary budget (hidden from candidates)

Table 4: events.csv

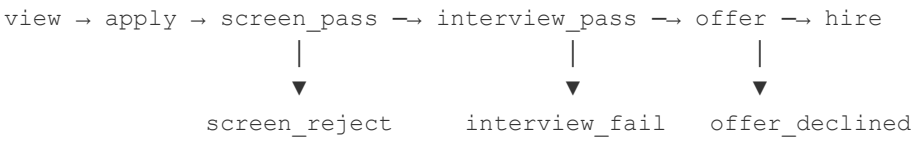
Rows: ~200,000 • **Unit of observation:** One action/event on the platform

Column	Type	Description
event_id	string	Primary key
candidate_id	string	Foreign key → candidates.candidate_id
job_id	string	Foreign key → jobs.job_id
event_type	string	Type of event (see Event Flow)

timestamp	string	When the event occurred (ISO format)
relevance_score	float	Platform's match score for candidate-job pair (0–1)
offered_salary	float	Salary offered (only for offer and hire events)
decline_reason	string	Reason for declining (only for offer_declined events)

Event Flow (Recruiting Funnel)

Events represent the stages of the recruiting process:



Event Type Definitions

event_type	Description	Next Possible Events
view	Candidate viewed the job posting	apply (or nothing)
apply	Candidate submitted an application	screen_pass, screen_reject
screen_pass	Application passed initial screening	interview_pass, interview_fail
screen_reject	Application rejected at screening	(terminal)
interview_pass	Candidate passed interview stage	offer, no_offer
interview_fail	Candidate failed interview	(terminal)
offer	Employer extended a job offer	hire, offer_declined
no_offer	Employer chose not to extend offer	(terminal)
hire	Candidate accepted the offer	(terminal — success!)
offer_declined	Candidate rejected the offer	(terminal)

Event Counts (for reference)

event_type	Count
view	134,789
apply	36,605
screen_reject	20,330
screen_pass	2,643
interview_pass	2,011
offer	1,405

offer_declined	751
hire	654
interview_fail	632
no_offer	606

Special Column Notes

offered_salary — Only has values for:

- `offer` events (the salary being offered)
- `hire` events (the salary accepted)
- All other event types: NULL

decline_reason — Only has values for `offer_declined` events:

- `salary_mismatch` — Candidate declined because salary was too low
- `other` — Candidate declined for other reasons

Common Analysis Patterns

Counting events by type

```
events['event_type'].value_counts()
```

Building the funnel

```
stages = ['view', 'apply', 'screen_pass', 'interview_pass', 'offer', 'hire']
for stage in stages:
    count = len(events[events['event_type'] == stage])
    print(f"{stage}: {count}")
```

Applications per candidate

```
apps = events[events['event_type'] == 'apply'].groupby('candidate_id').size()
```

Applications per job

```
apps = events[events['event_type'] == 'apply'].groupby('job_id').size()
```

Note: Jobs with zero applications won't appear in this groupby. To include them:

```
all_jobs = jobs['job_id']
apps = events[events['event_type']=='apply'].groupby('job_id').size()
       .reindex(all_jobs, fill_value=0)
```

Merging tables

```
# Add candidate info to events
events_with_candidates = events.merge(candidates, on='candidate_id')

# Add job info to events
```

```
events_with_jobs = events.merge(jobs, on='job_id')

# Add job AND candidate info
events_full = events.merge(candidates, on='candidate_id')
              .merge(jobs, on='job_id')
```

Offer acceptance analysis

```
offers = events[events['event_type'] == 'offer']
hires = events[events['event_type'] == 'hire']
declined = events[events['event_type'] == 'offer_declined']
acceptance_rate = len(hires) / len(offers)
```

Candidates with No Activity

Not all candidates in `candidates.csv` appear in `events.csv`. To find inactive candidates:

```
active_ids = set(events['candidate_id'].unique())
all_ids = set(candidates['candidate_id'])
inactive_ids = all_ids - active_ids
print(f"Inactive candidates: {len(inactive_ids)}")
```