Hong Kong Job Fair Aggregation System: Data Collection and Storage Strategy

This document outlines the comprehensive strategy for collecting, processing, storing, and integrating job fair and recruitment event information from various sources in Hong Kong. The system is designed to prioritize physical events with information in Traditional Chinese (ZH-HK) and to provide daily updates to a Google Calendar.

1. Database Schema Design

1.1 Core Tables

Events Table

- `event_id` (UUID, PK): Unique identifier for each event
- `event_name` (VARCHAR(255), NOT NULL): Name of the job fair/recruitment event
- `event_name_en` (VARCHAR(255)): English name of the event (if available)
- `event_name_zh` (VARCHAR(255)): Chinese name of the event (if available)
- `organizer_id` (UUID, FK): Reference to the organizer
- `venue_id` (UUID, FK): Reference to the venue
- `start_datetime` (TIMESTAMP, NOT NULL): Start date and time
- `end_datetime` (TIMESTAMP, NOT NULL): End date and time
- `description` (TEXT): Detailed description of the event
- `description_en` (TEXT): English description (if available)
- `description_zh` (TEXT): Chinese description (if available)
- `registration_method` (TEXT): How to register for the event
- `website_link` (VARCHAR(512)): Official event website
- `contact_email` (VARCHAR(255)): Contact email
- `contact_phone` (VARCHAR(50)): Contact phone number
- `language` (ENUM): Primary language of the event (ZH-HK, EN, BOTH)
- `target_audience` (TEXT): Target audience description
- `cost` (DECIMAL(10,2)): Cost to attend (0 for free events)
- `is_physical` (BOOLEAN, DEFAULT TRUE): Whether it's a physical event
- `is_virtual` (BOOLEAN, DEFAULT FALSE): Whether it has virtual components
- `google_maps_link` (VARCHAR(512)): Google Maps link to the venue
- `source_id` (UUID, FK): Reference to the information source
- `source_event_id` (VARCHAR(255)): Original ID from the source (if available)
- `created_at` (TIMESTAMP): Record creation timestamp
- `updated_at` (TIMESTAMP): Record last update timestamp
- `verified` (BOOLEAN, DEFAULT FALSE): Whether the information has been verified
- `status` (ENUM): Status of the event (UPCOMING, ONGOING, COMPLETED, CANCELLED)

Organizers Table

- `organizer_id` (UUID, PK): Unique identifier for each organizer
- `name` (VARCHAR(255), NOT NULL): Name of the organizer
- `name_en` (VARCHAR(255)): English name
- `name_zh` (VARCHAR(255)): Chinese name
- `description` (TEXT): Description of the organizer
- `website` (VARCHAR(512)): Organizer's website
- `type` (ENUM): Type of organization (GOVERNMENT, PRIVATE, UNIVERSITY,

NGO, OTHER)
- `created_at` (TIMESTAMP): Record creation timestamp
- `updated_at` (TIMESTAMP): Record last update timestamp

Venues Table

- `venue_id` (UUID, PK): Unique identifier for each venue
- `name` (VARCHAR(255), NOT NULL): Name of the venue
- `name_en` (VARCHAR(255)): English name
- `name_zh` (VARCHAR(255)): Chinese name
- `address` (TEXT, NOT NULL): Full address
- `address_en` (TEXT): English address
- `address_zh` (TEXT): Chinese address
- `district` (VARCHAR(100)): District in Hong Kong
- `latitude` (DECIMAL(10,8)): Geographical latitude
- `longitude` (DECIMAL(11,8)): Geographical longitude
- `transport_info` (TEXT): Transportation information
- `created_at` (TIMESTAMP): Record creation timestamp
- `updated_at` (TIMESTAMP): Record last update timestamp

Exhibitors Table

- `exhibitor_id` (UUID, PK): Unique identifier for each exhibitor
- `name` (VARCHAR(255), NOT NULL): Name of the exhibitor
- `name_en` (VARCHAR(255)): English name
- `name_zh` (VARCHAR(255)): Chinese name
- `description` (TEXT): Description of the exhibitor
- `website` (VARCHAR(512)): Exhibitor's website
- `industry` (VARCHAR(255)): Industry category
- `created_at` (TIMESTAMP): Record creation timestamp
- `updated_at` (TIMESTAMP): Record last update timestamp

Event_Exhibitors (Junction Table)

- `event_id` (UUID, FK, PK): Reference to the event
- `exhibitor_id` (UUID, FK, PK): Reference to the exhibitor
- `booth_number` (VARCHAR(50)): Booth number at the event (if available)
- `created_at` (TIMESTAMP): Record creation timestamp

Images Table

- `image_id` (UUID, PK): Unique identifier for each image
- `event_id` (UUID, FK): Reference to the event
- `url` (VARCHAR(512), NOT NULL): URL of the image
- `type` (ENUM): Type of image (POSTER, FLOOR_PLAN, BANNER, OTHER)
- `alt_text` (VARCHAR(255)): Alternative text for the image
- `created_at` (TIMESTAMP): Record creation timestamp

Sources Table

- `source_id` (UUID, PK): Unique identifier for each source
- `name` (VARCHAR(255), NOT NULL): Name of the source
- `url` (VARCHAR(512), NOT NULL): URL of the source
- `type` (ENUM): Type of source (GOVERNMENT, JOB_PORTAL, UNIVERSITY, EXHIBITION, OTHER)
- `priority` (ENUM): Priority level (PRIMARY, SECONDARY)
- `check_frequency` (ENUM): How often to check (DAILY, WEEKLY)
- `language` (ENUM): Primary language of the source (ZH-HK, EN, BOTH)
- `last_checked` (TIMESTAMP): When the source was last checked

- `created_at` (TIMESTAMP): Record creation timestamp
- `updated_at` (TIMESTAMP): Record last update timestamp

Past_Events Table

- `past_event_id` (UUID, PK): Unique identifier for past event
- `event_id` (UUID, FK): Reference to the original event
- `year` (INTEGER, NOT NULL): Year the past event was held
- `attendance` (INTEGER): Estimated attendance
- `exhibitor_count` (INTEGER): Number of exhibitors
- `notes` (TEXT): Additional notes about the past event
- `created_at` (TIMESTAMP): Record creation timestamp
- `updated_at` (TIMESTAMP): Record last update timestamp

1.2 Indexes and Constraints

- Primary keys on all ID fields
- Foreign key constraints with cascading updates and restricted deletes
- Composite indexes on frequently queried fields:
    - (start_datetime, end_datetime, status) for date range queries
    - (is_physical, district) for location-based queries
    - (language, target_audience) for audience targeting
    - (source_id, source_event_id) for deduplication

1.3 Data Types and Relationships

- One-to-many relationships:
    - Organizer to Events
    - Venue to Events
    - Source to Events
- Many-to-many relationships:
    - Events to Exhibitors (through Event_Exhibitors junction table)
- One-to-many relationship:
    - Events to Images
    - Events to Past_Events

2. Data Collection Methodology

2.1 Source-Specific Approaches

2.1.1 Primary Sources (Daily Check)

Hong Kong Labour Department

- **Method**: Web scraping with BeautifulSoup/Selenium
- **URL**: https://www2.jobs.gov.hk/0/tc/information/Epem/Vacancy/
- **Language**: Prioritize ZH-HK data, collect EN if available
- **Frequency**: Daily at 8:00 AM HKT
- **Extraction approach**:
    - Navigate to the job fair listing page
    - Extract event cards with dates, locations, and links
    - Follow links to detailed pages for complete information
    - Parse HTML tables for exhibitor lists

JobsDB Hong Kong

- **Method**: Web scraping with Selenium (handles dynamic content)
- **URL**: https://hk.jobsdb.com/招聘日-recruitment-day-jobs

- **Language**: Collect both ZH-HK and EN data
- **Frequency**: Daily at 9:00 AM HKT
- **Extraction approach**:
    - Navigate to recruitment day listings
    - Scroll to load all content (dynamic loading)
    - Extract event cards and details
    - Follow links to company pages for additional information

Hong Kong Trade Development Council

- **Method**: Web scraping with BeautifulSoup
- **URL**: https://www.hktdc.com/event/hkeducationexpo/tc
- **Language**: Prioritize ZH-HK data, collect EN if available
- **Frequency**: Daily at 10:00 AM HKT (more frequent during expo periods)
- **Extraction approach**:
    - Extract main expo information
    - Parse event schedule and exhibitor list
    - Collect detailed information about special zones and activities

2.1.2 Secondary Sources (Weekly Check)

University Career Centers

- **Method**: Web scraping with BeautifulSoup/Selenium
- **URLs**: Various university career center websites
- **Language**: Collect both EN and ZH-HK data
- **Frequency**: Weekly on Monday at 11:00 AM HKT
- **Extraction approach**:
    - Navigate to event calendars and listings
    - Extract public events (filter out student-only events if possible)
    - Follow links to detailed pages
    - Handle login requirements by focusing on publicly accessible information

Other Secondary Sources

- **Method**: Web scraping with appropriate tools based on site structure
- **URLs**: Various as listed in the sources document
- **Language**: Prioritize ZH-HK data where available
- **Frequency**: Weekly on Tuesday at 11:00 AM HKT
- **Extraction approach**:
    - Customize extraction for each source’s unique structure
    - Focus on upcoming events within the next 30 days
    - Extract as much detail as available

2.2 Language Handling

- **Priority**: Traditional Chinese (ZH-HK) content
- **Approach**:
    - Store both ZH-HK and EN versions when available
    - Use language detection to identify content language
    - For sources with only EN content, store as is and flag for translation
    - For mixed language content, separate and store in appropriate fields
    - Use OpenCC for simplified to traditional Chinese conversion when needed

2.3 Data Format Handling

- **HTML Content**:
    - Strip HTML tags while preserving formatting

    - Extract structured data from tables and lists
    - Convert formatted content to markdown for storage
- **PDF Documents**:
    - Use PyPDF2/pdfplumber for text extraction
    - OCR with Tesseract for image-based PDFs
    - Extract tables using Camelot/Tabula
- **Images**:
    - Store URLs rather than binary data
    - Extract text from images using OCR when necessary
    - Generate thumbnails for preview purposes

2.4 Scheduling and Automation

- **Primary Sources**:
    - Daily checks scheduled via cron jobs
    - Staggered timing to distribute load
    - Retry mechanism for failed attempts (3 retries with exponential backoff)
- **Secondary Sources**:
    - Weekly checks scheduled via cron jobs
    - Different days for different source categories
    - Lower priority in queue

3. Data Processing Pipeline

3.1 Pipeline Architecture

1. **Data Collection**: Source-specific scrapers/extractors
2. **Raw Data Storage**: JSON files in staging area
3. **Preprocessing**: Cleaning, normalization, language detection
4. **Entity Extraction**: Identify events, organizers, venues, exhibitors
5. **Deduplication**: Match against existing records
6. **Enrichment**: Add missing information from secondary sources
7. **Validation**: Check for required fields and data quality
8. **Database Storage**: Insert/update records in database
9. **Post-processing**: Generate notifications, update calendar
10. **Logging and Monitoring**: Track pipeline performance

3.2 Deduplication Methodology

- **Primary Key**: Generate a composite key based on:
    - Normalized event name (lowercase, remove punctuation)
    - Start date (YYYY-MM-DD)
    - Venue name (normalized)
    - Organizer name (normalized)
- **Fuzzy Matching**:
    - Use Levenshtein distance for name similarity (threshold: 85%)
    - Date proximity check (same day or adjacent days)
    - Location similarity check
- **Resolution Strategy**:
    - If exact match found: Update existing record
    - If fuzzy match found: Flag for manual review
    - If no match found: Create new record
- **Source Priority**:
    - When conflicts exist, prefer information from higher priority sources
    - Maintain source attribution for all data points

3.3 Verification and Cross-checking

- **Cross-source Verification**:
  - Compare event details across multiple sources
  - Flag discrepancies for manual review
  - Confidence score based on number of confirming sources
- **Temporal Verification**:
  - Check for logical date sequences (start before end)
  - Flag unusual timing patterns (very short or very long events)
- **Spatial Verification**:
  - Validate addresses against Hong Kong postal database
  - Verify venue capacity against expected attendance
- **Contact Verification**:
  - Validate email formats and domain existence
  - Validate phone number formats for Hong Kong

3.4 Data Enrichment Strategies

- **Venue Enrichment**:
  - Geocode addresses to obtain lat/long coordinates
  - Generate Google Maps links
  - Add transportation information from Google Places API
- **Organizer Enrichment**:
  - Link to company profiles from JobsDB or LinkedIn
  - Add industry classification
  - Include historical event information
- **Event Enrichment**:
  - Add weather forecast for outdoor events
  - Include historical attendance data if available
  - Link to related events (same organizer, venue, or theme)
- **Language Enrichment**:
  - Machine translation for missing language versions (marked as auto-translated)
  - Terminology standardization for industry-specific terms

3.5 Error Handling and Logging

- **Error Levels**:
  - Critical: Pipeline failure, database connection issues
  - Error: Source extraction failure, data validation failure
  - Warning: Missing non-critical fields, potential duplicates
  - Info: Normal operation logs
- **Logging Strategy**:
  - Structured JSON logs with timestamp, source, operation, status
  - Rotating log files with 30-day retention
  - Critical errors trigger email notifications
- **Recovery Mechanisms**:
  - Automatic retry for transient failures
  - Circuit breaker pattern for persistent source issues
  - Transaction rollback for database integrity
- **Monitoring Dashboard**:
  - Daily success/failure rates by source
  - New event count trends
  - Data quality metrics
  - Pipeline performance metrics

4. Storage and Backup Strategy

4.1 Primary Storage Solution

- **Database**: PostgreSQL 14+
  - Relational structure for complex relationships
  - Strong data integrity constraints
  - Support for JSON fields for semi-structured data
  - Full-text search capabilities for Chinese and English
- **Hosting**:
  - Managed PostgreSQL service (AWS RDS or equivalent)
  - Multi-AZ deployment for high availability
  - Performance tier: db.t4g.medium (minimum)
  - Storage: 100GB SSD with auto-scaling
- **Connection Pooling**:
  - PgBouncer for efficient connection management
  - Connection limits based on workload patterns

4.2 Backup Methodology

- **Automated Backups**:
  - Daily full database backups at 3:00 AM HKT
  - Point-in-time recovery enabled (5-minute intervals)
  - Transaction log archiving
- **Backup Storage**:
  - Primary: Cloud storage (S3 or equivalent)
  - Secondary: Different region/provider for disaster recovery
- **Backup Testing**:
  - Monthly restoration test to verify backup integrity
  - Quarterly disaster recovery simulation
- **Raw Data Preservation**:
  - Store original scraped data as JSON files
  - Daily compression and archiving
  - Separate storage from processed data

4.3 Data Retention Policies

- **Event Data**:
  - Active events: Indefinite retention
  - Past events: 5-year retention in primary database
  - Archived events: Moved to cold storage after 5 years
- **Raw Scraped Data**:
  - 90-day retention in hot storage
  - 1-year retention in cold storage
- **Logs and Metrics**:
  - Operational logs: 30-day retention
  - Error logs: 90-day retention
  - Performance metrics: 1-year retention
- **User Activity Data**:
  - Search history: 60-day retention
  - User preferences: Until explicitly changed

5. Integration Points

5.1 Google Calendar Integration

- **Integration Method**: Google Calendar API
- **Authentication**: OAuth 2.0 with service account
- **Target Calendar**: leekaben@gmail.com
- **Event Creation Logic**:
  - Create new calendar events for new job fairs
  - Update existing events when details change

    - Cancel events when job fairs are cancelled
- **Event Content**:
    - Title: Event name (ZH-HK primary, EN in parentheses if available)
    - Date/Time: Start and end times with proper timezone (HKT)
    - Location: Venue name and address with Google Maps link
    - Description: Formatted event details including:
        - Organizer information
        - Registration method
        - Exhibitor list (truncated if too long)
        - Website link
        - Contact information
        - Target audience
        - Cost information
        - Transportation details
    - Attachments: Link to event images
- **Update Frequency**:
    - Real-time updates for new events
    - Batch updates for modified events (twice daily)
- **Error Handling**:
    - Retry logic for API rate limits
    - Notification for persistent failures
    - Manual override capability

5.2 Monthly Report Generation

- **Report Format**: Excel (XLSX) and CSV
- **Generation Schedule**: 1st day of each month at 6:00 AM HKT
- **Report Content**:
    - Summary statistics (total events, by district, by industry)
    - Complete event listings for the upcoming month
    - Historical comparison with previous months/years
    - Source distribution analysis
- **Delivery Method**:
    - Stored in `/exports` directory
    - Email notification with download link
    - Direct email attachment option
- **Customization Options**:
    - Filterable by district, date range, industry
    - Sortable by various criteria
    - Pivot table functionality

5.3 API for Future Extensions

- **API Type**: RESTful API with JSON responses
- **Authentication**: API key and JWT for secure access
- **Endpoints**:
    - `/events`: List all events with filtering options
    - `/events/{id}`: Detailed information for a specific event
    - `/venues`: List all venues
    - `/organizers`: List all organizers
    - `/exhibitors`: List all exhibitors
- **Rate Limiting**: 1000 requests per day per API key
- **Documentation**: OpenAPI/Swagger specification
- **Versioning**: Semantic versioning (v1, v2, etc.)

6. Implementation Timeline and Resources

6.1 Development Phases

1. **Phase 1 (Week 1-2)**: Database setup and schema implementation
2. **Phase 2 (Week 3-4)**: Primary source scrapers development
3. **Phase 3 (Week 5-6)**: Secondary source scrapers development
4. **Phase 4 (Week 7-8)**: Processing pipeline implementation
5. **Phase 5 (Week 9-10)**: Google Calendar integration
6. **Phase 6 (Week 11-12)**: Report generation and testing

6.2 Resource Requirements

- **Development**: 1 backend developer, 1 data engineer
- **Infrastructure**: Cloud hosting (AWS/GCP/Azure)
- **External Services**:
  - Google Maps API for geocoding
  - Google Calendar API for event integration
  - Translation API for language enrichment (optional)
- **Monitoring**: Application monitoring and alerting system

6.3 Maintenance Plan

- **Daily Operations**:
  - Monitor scraper success rates
  - Review flagged duplicates and conflicts
  - Verify calendar synchronization
- **Weekly Operations**:
  - Update scraper patterns for any site changes
  - Review data quality metrics
  - Test backup restoration
- **Monthly Operations**:
  - Generate and distribute monthly reports
  - Review system performance
  - Implement minor improvements

7. Conclusion

This data collection and storage strategy provides a comprehensive framework for aggregating job fair information from various sources in Hong Kong. The system prioritizes Traditional Chinese (ZH-HK) content and physical events, with a focus on providing complete and accurate information for daily updates to Google Calendar. The modular design allows for future expansion and adaptation as new sources become available or existing sources change their formats.