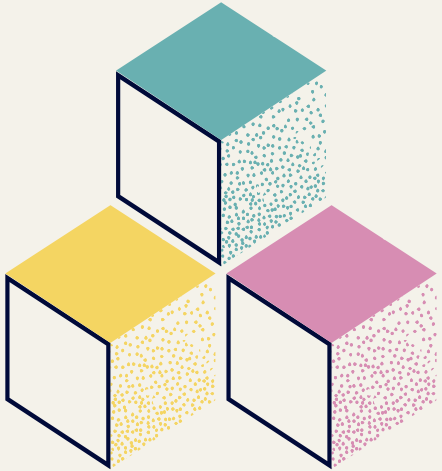


Recruitment Agency

Hire It!

GUKKA Student, Yuliya Yeusiyevich, 13.12.2022

Contents

Business Description

Model Description

01

Business Background

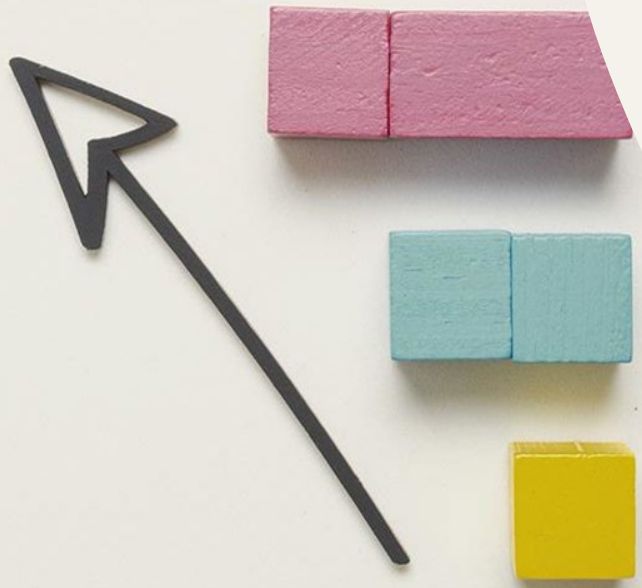
This section contains brief information
about the company

Recruitment agency Hire It! is committed to helping organizations to increase their productivity through better use of their human resources and to enable companies to achieve real competitive advantages. Our skills and expertise cover a wide range of HR activities and initiatives. We work across all industry sectors, employing a simple yet effective approach to each project.

02

Current Situation

This section contains information about the current situation with data management in the company as well associated problems and challenges



Currently, our company uses the traditional methods of tracking the recruitment process with **Excel** or **Google Sheets** spreadsheets include creating workbooks or downloading templates from the Internet.

In the near future our organization will begin to **grow** and our processes need to be able to handle this evolution and if our systems don’t scale up they’re pretty much dead in the water.

Problems

No Customer Relationship Management Software

No Applicant Tracking System in place

No Team Collaboration Tools

No Reports and Analytics

No Resume Parser

Lack of Scalability

Poor Data Integrity

Faulty Version Control

Erratic Disaster Recovery

Limited Security and Compliance

03

Solution

This section contains solution description: benefits of the **implementing a database**

- Improved data sharing and data security
- Consistent, reliable data
- Effective data integration
- Increased productivity
- Data that complies with privacy regulations
- Better decision-making

04

Definitions & Acronyms

Acronym	Definition
PK	primary key
FK	foreign key
AI	auto-increment
NN	not null
N	null
U	unique
1:1	one-to-one relationship
1:M	one-to-many relationship
M:1	many-to-one relationship
DB	database

05

Logical Scheme

This section contains ERD and description of the step-by-step modelling process

Hiring step-by-step

- Companies contact recruitment agencies to hire on their behalf. In some cases, companies recruit employees directly.
- The person responsible for recruitment starts the recruiting process. This process can have multiple steps, such as the initial screening, the first and follow-up interviews, the actual hiring decision, etc.
- Once the recruiters have agreed on a particular process – the vacancy is advertised on various platforms.
- Applicants start applying for the job.
- The applicants are shortlisted and invited to an initial interview.
- Applicants are evaluated on the basis of tests and interviews.
- The hiring decision is made.

DB modelling step-by-step

- 5 entities: selection processes, vacancies, applicants, interviews, recruiters.
- The **processes** table will store information about each recruitment process. Since one process can have multiple stages and one stage can be part of many processes, we need a bridge table – **process_stage**. The **stage** table contains information about the steps followed throughout that recruitment process.
- The **vacancies** table will store all the information related to the job(s) we're recruiting for. The **categories** table will broadly describe the type of job. We could expect to see job categories like IT, management, finance, education, etc. The **positions** table will contain the actual job title. Since one title can be advertised for multiple jobs (e.g. IT Manager, Sales Manager), we'll create a separate table for job positions. The **companies** table stores information about all the company who have ever used this database as part of their hiring process. The **contract_types** table is self-explanatory.

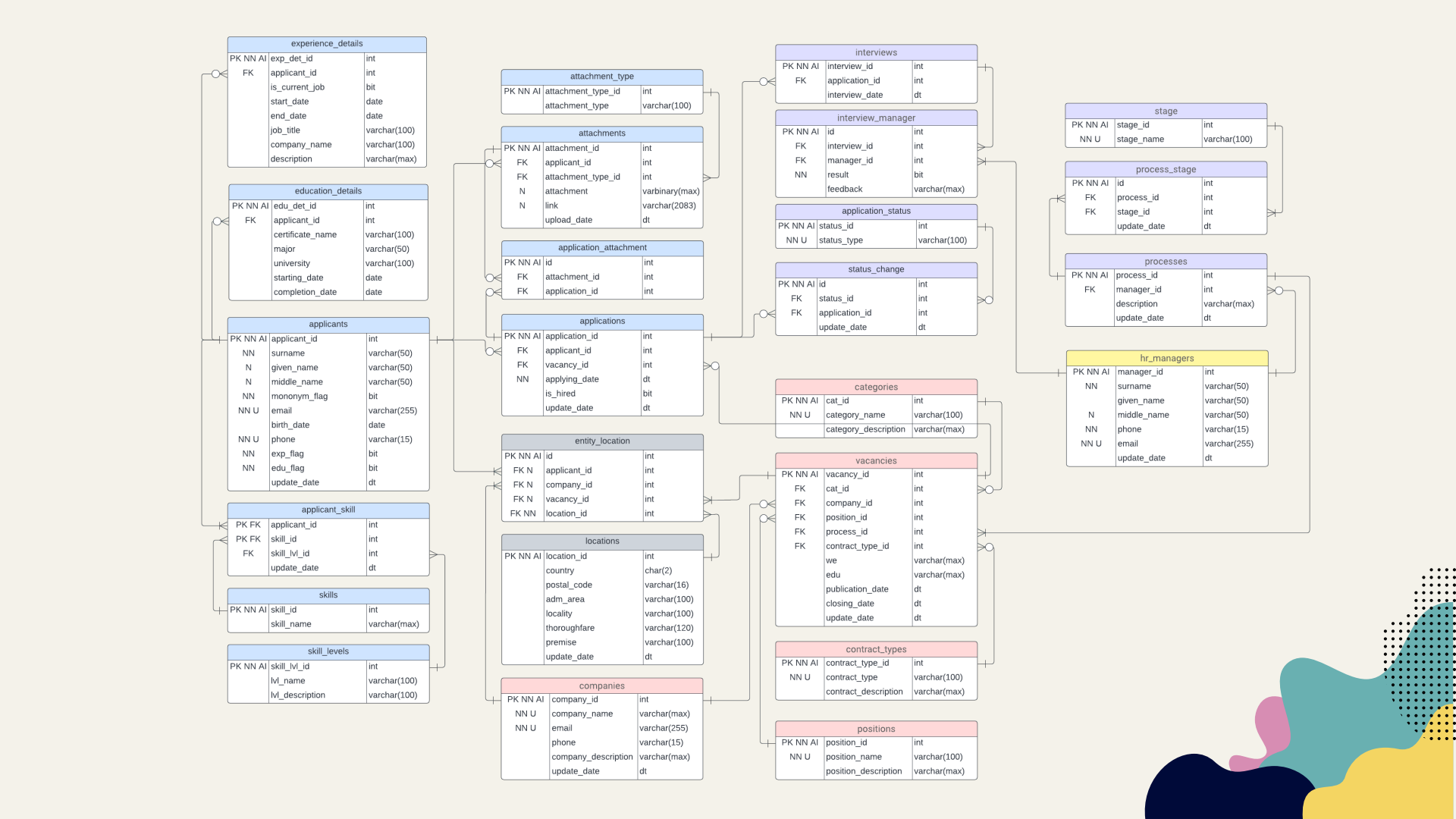
- The third part of the schema consists of the tables that store information about job applicants, their applications, and any attachments that come with the applications. The **applicants** table stores applicants' personal information, such as their first name, last name, email, phone number, etc. As our agency moves to the international level, we will add mononymous flag for mononymous people (particularly in countries like India and Indonesia) and change name fields to *surname*, *given name* and *middle name* (in many Asian countries the order of names differs: last name – first name). Two additional flags: *exp_flag* and *edu_flag* indicate whether an applicant has some work experience or education. The **applications** table contains information for each application and also the final decision about each application – hired\not hired. Since there can be multiple applications for each job but each application is only for one job, there will be a one-to-many relationship between the **vacancies** and **applications** tables.

- Similarly, one applicant can submit multiple applications (i.e. for different jobs), but each application is from only one participant; we've implemented another one-to-many relationship between the **applicants** and **applications** tables to handle this. The **attachments** table manages the supporting documents that applicants may attach to their application. These can be CVs, resumes, letters of reference, cover letters, etc. (this info will be stored in the addition **attachment_type** table). Attachment table has a binary column named *attachment*, which will store the file in binary format. A link to the document may be stored in the *link* field. Both field are nullable. Neither is mandatory, and an applicant may choose to use either or both methods to add information to their application. One document can be attached to multiple applications, and one application can have multiple supporting documents. This means there is a many-to-many relationship so we create the bridge table **application_attachment**.

- Detailed information about applicant's education and experience will be stored in the **education_details** and **experience_details** tables with many-to-one relationships with the applicants table. The **education_details** table will store name of the certificate (as well as major and university for the higher education) and the start\completion dates. The **experience_details** table will store *is_current_job* flag, name of the company, position and start\end dates. All applicant's skills will be stored separately – in the **skills** table with many-to-many relationship with the applicants table (so an additional bridge table **applicant_skill** will be created). All skills will have a level system which will be stored in the **skill_levels** table. The level information can be found in the bridge table applicant_skill.

- The **interviews** table stores basic information for each interview. One interview can be associated with only one application. On the other hand, one application can have multiple interviews. Therefore, a one-to-many relationship exists between the **applications** and **interviews** table. One interview can be conducted by multiple managers, and one manager can take multiple interviews. It's another many-to-many relationship, so we'll created the bridge table **interview_manager**.
- An application can go through multiple stages during the hiring process, e.g. not submitted, under review, etc. The **application_status** table will be used to store such information. The **status_change** table will contain status changes for all submitted applications. This table can be handy if you want to analyze the processing time for each stage of different applications.

- The fourth part of our recruitment model stores information about hr managers, application statuses, and application evaluations. The **hr_managers** table stores each recruiter's first, last name, phone, email and unique id number.
- The last part of our recruitment model will store general information about location of the applicant, company or vacancy (in the **locations** table). Based on the principle that one entity can have multiple addresses, which is usually the case (home, business, etc.), and that one address can be shared, we will create the lookup table **entity_location.** The vacancies, companies and applicants tables will have a one-to-many relationship with this bridge table.


experience_details
PK NN AI exp_det_id int
FK applicant_id int
is_current_job bit
start_date date
end_date date
job_title varchar(100)
company_name varchar(100)
description varchar(max)
education_details
PK NN AI edu_det_id int
FK applicant_id int
certificate_name varchar(100)
major varchar(50)
university varchar(100)
starting_date date
completion_date date
applicants
PK NN AI applicant_id int
NN surname varchar(50)
N given_name varchar(50)
N middle_name varchar(50)
NN mononym_flag bit
NN U email varchar(255)
birth_date date
NN U phone varchar(15)
NN exp_flag bit
NN edu_flag bit
update_date dt
applicant_skill
PK FK applicant_id int
PK FK skill_id int
FK skill_lvl_id int
update_date dt
skills
PK NN AI skill_id int
skill_name varchar(max)
skill_levels
PK NN AI skill_lvl_id int
lvl_name varchar(100)
lvl_description varchar(100)
attachment_type
PK NN AI attachment_type_id int
attachment_type varchar(100)
attachments
PK NN AI attachment_id int
FK applicant_id int
FK attachment_type_id int
N attachment varbinary(max)
N link varchar(2083)
upload_date dt
application_attachment
PK NN AI id int
FK attachment_id int
FK application_id int
applications
PK NN AI application_id int
FK applicant_id int
FK vacancy_id int
NN applying_date dt
is_hired bit
update_date dt
entity_location
PK NN AI id int
FK N applicant_id int
FK N company_id int
FK N vacancy_id int
FK NN location_id int
locations
PK NN AI location_id int
country char(2)
postal_code varchar(16)
adm_area varchar(100)
locality varchar(100)
thoroughfare varchar(120)
premise varchar(100)
update_date dt
companies
PK NN AI company_id int
NN U company_name varchar(max)
NN U email varchar(255)
phone varchar(15)
company_description varchar(max)
update_date dt
interviews
PK NN AI interview_id int
FK application_id int
interview_date dt
interview_manager
PK NN AI id int
FK interview_id int
FK manager_id int
NN result bit
feedback varchar(max)
application_status
PK NN AI status_id int
NN U status_type varchar(100)
status_change
PK NN AI id int
FK status_id int
FK application_id int
update_date dt
categories
PK NN AI cat_id int
NN U category_name varchar(100)
category_description varchar(max)
vacancies
PK NN AI vacancy_id int
FK cat_id int
FK company_id int
FK position_id int
FK process_id int
FK contract_type_id int
we varchar(max)
edu varchar(max)
publication_date dt
closing_date dt
update_date dt
contract_types
PK NN AI contract_type_id int
NN U contract_type varchar(100)
contract_description varchar(max)
positions
PK NN AI position_id int
NN U position_name varchar(100)
position_description varchar(max)
stage
PK NN AI stage_id int
NN U stage_name varchar(100)
process_stage
PK NN AI id int
FK process_id int
FK stage_id int
update_date dt
processes
PK NN AI process_id int
FK manager_id int
description varchar(max)
update_date dt
hr_managers
PK NN AI manager_id int
NN surname varchar(50)
given_name varchar(50)
N middle_name varchar(50)
NN phone varchar(15)
NN U email varchar(255)
update_date dt

06

Objects

This section contains DB objects: tables with descriptive information, relationships between them, table attributes, and examples with filled data

Table **applicants** stores applicants' personal information and allows for mononymous people.

Table Name	Field Name	Field Description	Data Type
applicants	applicant_id	PK, NN, AI, unique applicant's ID	int
	surname	NN, applicant's last name	varchar(50)
	given_name	N, applicant's first name	varchar(50)
	middle_name	N	varchar(50)
	mononym_flag	NN, flag, whether the user is a mononym	bit
	email	NN, U	varchar(255)
	birth_date		date
	phone	applicant's phone number	varchar(15)
	exp_flag	NN, flag, whether the user has a working experience	bit
	edu_flag	NN, flag, whether the user is an education	bit
	update_date		dt

Table **relations:** 1:M with the **applications, attachments**, **entity_location, education_details, experience_details, applicant_skill** tables**.**

applicant_id	surname	given_name	middle_name	mononym_flag	email	birth_date	phone	exp_flag	edu_flag	update_date
111	Potter	Harry	NULL	0	hp@gmail.com	01.01.1990	37529 77777 77	1	1	2022-12-12 9:30:30.260

Table **education_details** contains information on applicant's education.

Table Name	Field Name	Field Description	Data Type
education_details	edu_det_id	PK, NN, AI, unique details ID	int
	applicant_id	FK, reference to the **applicants** table	int
	certificate_name	name of the diploma\certificate	varchar(100)
	major	degree	varchar(50)
	university	name of the university	varchar(100)
	starting_date		date
	completion_date		date

Table **relations:** M:1with the **applicants** table.

edu_det_id	applicant_id	certificate_name	major	university	starting_date	completion_date
1	111	MA in English Literature	master	Harvard University, Cambridge	2010-09-01	2014-06-30

Table **experience_details** contains information on applicant's working experience.

Table Name	Field Name	Field Description	Data Type
experience_details	exp_det_id	PK, NN, AI, unique details ID	int
	applicant_id	FK, reference to the **applicants** table	int
	is_current_job	flag, whether it's the current job	bit
	start_date		date
	end_date		date
	job_title		varchar(100)
	company_name		varchar(100)
	description		varchar(max)

Table **relations:** M:1with the **applicants** table.

exp_det_id	applicant_id	is_current_job	start_date	end_date	job_title	company_name	description
1	111	0	2014-07-01	2018-07-01	teacher	Trinity Episcopal School	Responsible for preparing lesson plans...

The bridge table **applicant_skill** connects **applicants** and **skills** tables.

Table Name	Field Name	Field Description	Data Type
applicant_skill	applicant_id	PK, FK, reference to the **applicants** table	int
	skill_id	PK, FK, reference to the **skills** table	int
	skill_lvl_id	FK, reference to the **skill_levels** table	bit
	update_date		dt

Table **relations:** M:1with the **applicants, skills, skill_levels** tables.

applicant_id	skill_id	skil_lvl_id	update_date
111	23	3	2022-12-12 9:30:30.260

Skills and skill_levels tables are self-explanatory.

Table Name	Field Name	Field Description	Data Type
skills	skill_id	PK, NN, AI unique skill ID	int
	skill_name		varchar(max)

Table **relations** (skills)**:** 1:M with the **applicants** table.

Table Name	Field Name	Field Description	Data Type
skill_levels	skill_lvl_id	PK, NN, AI unique level ID	int
	lvl_name		varchar(100)
	lvl_description		varchar(max)

Table **relations** (skill_levels)**:** 1:M with the **applicant_skill** table.

skill_id	skill_name
1	English language

skill_lvl_id	lvl_name	lvl_description
1	novice	Has minimal or textbook knowledge without connecting it...

Table **applications** contains information on each application.

Table Name	Field Name	Field Description	Data Type
applications	application_id	PK, NN, AI, unique application ID	int
	applicant_id	FK, reference to the **applicants** table	int
	vacancy_id	FK, reference to the **vacancies** table	int
	applying_date		dt
	is_hired	final decision about an application	bit
	update_date		dt

Table **relations:** 1:M with the bridge table **application_attachment, status_change**, M:1 with **applicants, vacancies**.

application_id	applicant_id	vacancy_id	applying_date	is hired	update_date
345	111	568	2022-12-12 10:30:30.260	1	2022-12-12 9:30:30.260

Application_attachment is a bridge table between **applications** and **attachments** tables.

Table Name	Field Name	Field Description	Data Type
application_attachment	id	PK, NN, AI, unique row ID	int
	attachment_id	FK, reference to the **attachments** table	int
	application_id	FK, reference to the **application** table	int

Table **relations:** M:1 with bridge tables **application** and **attachments.**

id	attachment_id	application_id
123	876	345

Table **attachments** contains information on each application's attachment and an attachment itself (or it's URL).

Table Name	Field Name	Field Description	Data Type
attachments	attachment_id	PK, NN, AI, unique attachment ID	int
	applicant_id	FK, reference to the **applicants** table	int
	attachment_type_id	FK, reference to the **attachment_type** table	varchar(100)
	attachment	N, file in binary format	varbinary(max)
	link	N, url, link to the document	varchar(2083)
	upload_date		dt

Table **relations:** 1:M with the bridge table **application_attachment,** M:1 with **attachment_type, applicants** tables.

attachment_id	applicant_id	attachment_type_id	attachment	link	upload_date
876	11	444	NULL	Curriculum vitae - Wikipedia	2022-12-12 12:30:30.260

Table **attachment_type** contains information on attachments' type.

Table Name	Field Name	Field Description	Data Type
attachment_type	attachment_type_id	PK, NN, AI, unique attachment type ID	int
	attachment_type	possible options: resume, CV, cover letter etc.	varchar(100)

Table **relations:** 1:M with the bridge table **attachments.**

attachment_type_id	attachment_type
444	CV

Table **locations** stores information about location of a company or an applicant.

Table Name	Field Name	Field Description	Data Type
locations	location_id	PK, NN, AI, unique location ID	int
	country	an alpha-2 country code	char(2)
	postal_code	Postal code / ZIP Code	varchar(16)
	adm_area	State / Province / Region (ISO code when available)	varchar(100)
	locality	City / Town	varchar(100)
	thoroughfare	Street address	varchar(100)
	premise	Apartment, Suite, Box number, etc.	varchar(100)
	update_date		dt

Table **relations:** 1:M with the bridge table **entity_location.**

location_id	country	postal_code	adm_area	locality	thoroughfare	premise	update_date
8889	US	94578	CA	San Leandro	Station Street	1379	2022-12-12 12:30:30.260

Entity_location is a bridge table between location and **applicants\companies\vacancies** tables.

Table Name	Field Name	Field Description	Data Type
entity_location	id	PK, NN, AI, unique row ID	int
	location_id	FK, NN, reference to the **locations** table	int
	company_id	FK, N, reference to the **companies** table	int
	applicant_id	FK, N, reference to the **applicants** table	int
	vacancy_id	FK, N, reference to the **vacancies** table	int

Table **relations:** M:1 with the **location, applicants and companies** tables.

id	location_id	company_id	applicant_id
346586	8889	349687	NULL

Table **companies** contains information on organizations that contact recruitment agency to hire on their behalf. In some cases, companies recruit employees directly.

Table Name	Field Name	Field Description	Data Type
companies	company_id	PK, NN, AI, unique company ID	int
	company_name	NN, U	varchar(max)
	email	NN, U, contact person's email	varchar(255)
	phone	phone number of the contact person in the company	varchar(15)
	company_description		varchar(max)
	update_date		dt

Table **relations:** 1:M with the bridge table **entity_location** and **vacancies** table.

company_id	company_name	email	phone	company_description	update_date
34655686	Trophy Fitness Uptown	xyz@gmail.com	375441234567	Our mission is to provide our members at all fitness levels with an enjoyable...	2022-12-12 12:30:30.260

Table **vacancies** contains the actual job description. This table has many foreign keys, which means that it can be used to look up the category, position, platform, the hiring organization, and the recruitment process related to that vacancy.

Table Name	Field Name	Field Description	Data Type
vacancies	vacancy_id	PK, NN, AI, unique vacancy ID	int
	cat_id	FK, reference to the **categories** table	int
	company_id	FK, reference to the **companies** table	int
	position_id	FK, reference to the **positions** table	int
	process_id	FK, reference to the **processes** table	int
	contract_type_id	FK, reference to the **contract_types** table	int
	we	required working experience	varchar(max)
	edu	required level of education	varchar(max)
	publication_date		dt
	closing_date		dt
	update_date		dt

Table **relations:** 1:M with **applications**, **entity_location** tables. M:1 with **companies, categories, contract_types, processes** and **positions** tables.

vacancy_id	cat_id	company_id	position_id	process_id	contract_type_id	we	edu	pub_date	closing_date	update_date
15	2	34655686	14	22244	1	10+ years	BD, MD	2021-10-01 12:30:30.260	2022-05-01 10:30:30.260	2022-05-01 10:30:30.260

Table **contract_types** contains the information on type of the contract. All attributes are self-explanatory.

Table Name	Field Name	Field Description	Data Type
contract_types	contract_type_id	PK, NN, AI, unique type ID	int
	contract_type	NN, U, possible options: full-time, part-time, fixed-term etc.	varchar(100)
	contract_description		varchar(max)

Table **relations:** 1:M with **vacancies** table.

contract_type_id	contract_type	contract_description
1	fixed-term	Contract extended to employees who only work for a set amount of time or until they complete a specific task. Fixed-term contracts are...

Table **categories** broadly describes the type of job.

Table Name	Field Name	Field Description	Data Type
categories	cat_id	PK, NN, AI, unique category ID	int
	category_name	NN, U, possible options: IT, education, finance etc.	varchar(100)
	category_description		varchar(max)

Table **relations:** 1:M with **vacancies** table.

cat_id	category_name	category_description
1	IT	Any position that involves the implementation, support, maintenance, repair or protection of data or computer systems. Those involved in development...

Table **positions** describes the position of job.

Table Name	Field Name	Field Description	Data Type
positions	position_id	PK, NN, AI, unique position ID	int
	position_name	NN, U, possible options: Team Lead, Vice President etc.	varchar(100)
	position_description		varchar(max)

Table **relations:** 1:M with **vacancies** table.

position_id	position_name	position_description
34	Vice President	It is the second or third in command and supports the President by overseeing internal operations and...

Table **hr_managers** stores managers' information.

Table Name	Field Name	Field Description	Data Type
hr_managers	manager_id	PK, NN, AI, unique applicant's ID	int
	surname	NN, manager's last name	varchar(50)
	given_name	manager's first name	varchar(50)
	middle_name	N, manager's middle name	varchar(50)
	phone	manager's phone number	varchar(15)
	email		varchar(255)
	update_date		dt

Table **relations:** 1:M with the **interview_manager** and **processes** tables.

manager_id	surname	given_name	middle_name	phone	email	update_date
1	Smith	Sally	NULL	375291111213	janedoe@gmail.com	2022-12-12 12:30:30.260

The table **interviews** contains information on job interviews.

Table Name	Field Name	Field Description	Data Type
interviews	interview_id	PK, NN, AI, unique interview ID	int
	application_id	FK, reference to the **applications** table	int
	interview_date		dt

Table **relations:** 1:M with the bridge table **interview_manager,** M:1 with **applications.**

interview_id	application_id	interview_date
1	11	2022-10-12 12:30:30.260

The bridge table **interview_manager** connects **interviews** and **hr_managers** tables.

Table Name	Field Name	Field Description	Data Type
interview_manager	id	PK, NN, AI, unique row ID	int
	interview_id	FK, reference to the **interviews** table	int
	manager_id	FK, reference to the **hr_managers** table	int
	result	NN, result of the interview (hired\not hired)	bit
	feedback	manager's feedback	varchar(max)

Table **relations:** M:1 with the **interviews** and **hr_managers** tables.

id	interview_id	manager_id	result	feedback
1	11	111	0	The applicant came well-prepared for the interview and gave us the impression that...

The table **processes** contains information on each recruitment process.

Table Name	Field Name	Field Description	Data Type
processes	process_id	PK, NN, AI, unique process ID	int
	manager_id	FK, reference to the **hr_managers** table	int
	description		varchar(max)
	update_date		dt

Table **relations:** 1:M with **vacancies** and **process_stage** tables, M:1 with **hr_managers.**

process_id	manager_id	description	update_date
1000	11	Application from XYZ Company to hire 20 people for new marketing department...	2022-10-12 12:30:30.260

The bridge table **process_stage** connects **processes** and **stage** tables.

Table Name	Field Name	Field Description	Data Type
process_stage	id	PK, NN, AI, unique row ID	int
	process_id	FK, reference to the **processes** table	int
	stage_id	FK, reference to the **stage** table	int
	update_date		dt

Table **relations:** M:1 with the **processes** and **stage** tables.

id	process_id	stage_id	update_date
1	1000	2	2022-10-12 12:30:30.260

Table **stage** contains descriptive information on each stage of the process.

Table Name	Field Name	Field Description	Data Type
stage	stage_id	PK, NN, AI, unique stage ID	int
	stage_name	NN, U, options: initial screening, HR interview etc.	varchar(100)

Table **relations:** 1:M with **process_stage** table.

stage_id	stage_name
1	initial screening

The bridge table **status_change** is used to store information about changes in application's status.

Table Name	Field Name	Field Description	Data Type
status_change	id	PK, NN, AI, unique row ID	int
	status_id	FK, reference to the **application_status** table	int
	application_id	FK, reference to the **applications** table	int
	update_date		dt

Table **relations:** M:1 with the **application_status** and **applications** tables.

id	status_id	application_id	update_date
1	1000	2	2022-10-12 12:30:30.260

The bridge table **application_status** is used to store information about application's statuses.

Table Name	Field Name	Field Description	Data Type
application_status	status_id	PK, NN, AI, unique status ID	int
	status_type	NN, U options: not submitted, under review, waiting for decision etc.	int

Table **relations:** 1:M with **status_change** table.

status_id	status_type
1	not_submitted