-- STANDALONE TABLES:

-- Table: user\_account

CREATE TABLE user\_account (

id int NOT NULL AUTO\_INCREMENT,

password varchar(32) NOT NULL,

email varchar(128) NOT NULL UNIQUE,

first\_name varchar(128) NOT NULL,

last\_name varchar(128) NOT NULL,

dob date NOT NULL,

CONSTRAINT user\_account\_pk PRIMARY KEY (id)

);

-- Table: skill

CREATE TABLE skill (

id int NOT NULL AUTO\_INCREMENT,

skill\_name varchar(128) NOT NULL UNIQUE ,

CONSTRAINT skill\_pk PRIMARY KEY (id)

);

CREATE TABLE payment\_type (

id int NOT NULL,

type\_name varchar(128) NOT NULL UNIQUE,

CONSTRAINT payment\_type\_pk PRIMARY KEY (id)

);

-- FREELANCER AREA:

-- Table: freelancer

CREATE TABLE freelancer (

id int NOT NULL AUTO\_INCREMENT,

user\_account\_id int NOT NULL UNIQUE ,

registration\_date timestamp NOT NULL,

location varchar(255) NOT NULL,

overview text NOT NULL,

profile\_picture\_url varchar(128),

CONSTRAINT freelancer\_user\_account FOREIGN KEY (user\_account\_id) REFERENCES user\_account (id),

CONSTRAINT freelancer\_pk PRIMARY KEY (id)

);

-- Table: certification

CREATE TABLE certification (

id int NOT NULL AUTO\_INCREMENT,

freelancer\_id int NOT NULL,

certification\_name varchar(255) NOT NULL,

provider varchar(255) NOT NULL,

description text NOT NULL,

date\_earned date NOT NULL,

certification\_link text NULL,

CONSTRAINT certification\_freelancer FOREIGN KEY (freelancer\_id) REFERENCES freelancer (id),

CONSTRAINT certification\_pk PRIMARY KEY (id)

);

-- Table: has\_skill

CREATE TABLE has\_skill (

freelancer\_id int NOT NULL ,

skill\_id int NOT NULL ,

CONSTRAINT has\_skill\_freelancer FOREIGN KEY (freelancer\_id) REFERENCES freelancer (id), CONSTRAINT has\_skill\_skill FOREIGN KEY (skill\_id) REFERENCES skill (id) ,

CONSTRAINT has\_skill\_pk PRIMARY KEY (freelancer\_id, skill\_id)

);

-- CLIENT AREA:

-- Table: company

CREATE TABLE company (

id int NOT NULL AUTO\_INCREMENT,

company\_name varchar(128) NOT NULL,

company\_location varchar(255) NOT NULL,

CONSTRAINT company\_pk PRIMARY KEY (id)

);

CREATE TABLE hire\_manager (

id int NOT NULL AUTO\_INCREMENT,

user\_account\_id int NOT NULL UNIQUE ,

registration\_date timestamp NOT NULL,

location varchar(255) NOT NULL,

company\_id int NULL,

profile\_picture\_url varchar(128),

CONSTRAINT hire\_manager\_user\_account FOREIGN KEY (user\_account\_id) REFERENCES user\_account (id),

CONSTRAINT hire\_manager\_company FOREIGN KEY (company\_id) REFERENCES company(id),

CONSTRAINT hire\_manager\_pk PRIMARY KEY (id)

);

-- JOB POSTING AREA:

CREATE TABLE expected\_duration (

id int NOT NULL,

duration\_text varchar(255) NOT NULL UNIQUE ,

CONSTRAINT expected\_duration\_pk PRIMARY KEY (id)

);

CREATE TABLE complexity (

id int NOT NULL,

complexity\_text varchar(255) NOT NULL UNIQUE,

CONSTRAINT complexity\_pk PRIMARY KEY (id)

);

CREATE TABLE job (

id int NOT NULL AUTO\_INCREMENT,

hire\_manager\_id int NOT NULL,

expected\_duration\_id int NOT NULL,

complexity\_id int NOT NULL,

description text NOT NULL,

main\_skill\_id int NOT NULL,

payment\_type\_id int NOT NULL,

payment\_amount decimal(8,2) NOT NULL,

CONSTRAINT job\_hire\_manager FOREIGN KEY (hire\_manager\_id) REFERENCES hire\_manager (id),

CONSTRAINT job\_complexity FOREIGN KEY (complexity\_id) REFERENCES complexity (id),

CONSTRAINT job\_expected\_duration FOREIGN KEY (expected\_duration\_id) REFERENCES expected\_duration (id),

CONSTRAINT job\_skill FOREIGN KEY (main\_skill\_id) REFERENCES skill (id),

CONSTRAINT job\_payment\_type FOREIGN KEY (payment\_type\_id) REFERENCES payment\_type (id),

CONSTRAINT job\_pk PRIMARY KEY (id)

);

CREATE TABLE other\_skills (

job\_id int NOT NULL,

skill\_id int NOT NULL,

CONSTRAINT other\_skills\_job FOREIGN KEY (job\_id) REFERENCES job (id),

CONSTRAINT other\_skills\_skill FOREIGN KEY (skill\_id) REFERENCES skill (id),

CONSTRAINT other\_skills\_pk PRIMARY KEY (job\_id, skill\_id)

);

-- PROPOSAL AND CONTRACT:

CREATE TABLE proposal\_status\_catalog (

id int NOT NULL,

status\_name varchar(128) NOT NULL UNIQUE ,

CONSTRAINT proposal\_status\_catalog\_pk PRIMARY KEY (id)

);

CREATE TABLE proposal (

id int NOT NULL AUTO\_INCREMENT,

job\_id int NOT NULL,

freelancer\_id int NOT NULL,

proposal\_time timestamp NOT NULL,

payment\_type\_id int NOT NULL,

payment\_amount decimal(8,2) NOT NULL,

current\_proposal\_status int NOT NULL,

client\_grade int NULL,

client\_comment text NULL,

freelancer\_grade int NULL,

freelancer\_comment text NULL,

CONSTRAINT proposal\_freelancer FOREIGN KEY (freelancer\_id) REFERENCES freelancer (id),

CONSTRAINT proposal\_job FOREIGN KEY (job\_id) REFERENCES job (id),

CONSTRAINT proposal\_payment\_type FOREIGN KEY (payment\_type\_id) REFERENCES payment\_type (id),

CONSTRAINT proposal\_proposal\_status\_catalog FOREIGN KEY (current\_proposal\_status) REFERENCES proposal\_status\_catalog (id),

CONSTRAINT proposal\_pk PRIMARY KEY (id)

);

CREATE TABLE message (

id int NOT NULL AUTO\_INCREMENT,

proposal\_id int NOT NULL,

message\_time timestamp NOT NULL,

message\_text text NOT NULL,

message\_type int NOT NULL,

CONSTRAINT message\_proposal FOREIGN KEY (proposal\_id) REFERENCES proposal (id),

CONSTRAINT message\_pk PRIMARY KEY (id)

);

CREATE TABLE attachment (

id int NOT NULL AUTO\_INCREMENT,

message\_id int NOT NULL,

attachment\_link text NOT NULL,

CONSTRAINT attachment\_message FOREIGN KEY(message\_id) REFERENCES message(id),

CONSTRAINT attachment\_pk PRIMARY KEY (id)

);

CREATE TABLE contract (

id int NOT NULL AUTO\_INCREMENT,

proposal\_id int NOT NULL,

company\_id int NOT NULL,

freelancer\_id int NOT NULL,

payment\_type\_id int NOT NULL,

payment\_amount decimal(8,2) NOT NULL,

start\_time timestamp NOT NULL,

end\_time timestamp NULL,

CONSTRAINT contract\_proposal FOREIGN KEY (proposal\_id) REFERENCES proposal (id),

CONSTRAINT contract\_company FOREIGN KEY (company\_id) REFERENCES company (id),

CONSTRAINT contract\_freelancer FOREIGN KEY (freelancer\_id) REFERENCES freelancer (id),

CONSTRAINT contract\_payment\_type FOREIGN KEY (payment\_type\_id) REFERENCES payment\_type (id),

CONSTRAINT contract\_pk PRIMARY KEY (id)

);