UNIVERSITY OF
COPENHAGEN

5100-B4-4F23;Databases and Information Systems

Translation Agency "Easy Translate"

Client interaction with a web application

Laust Christian Blixencrone-Møller (sbh546)
Victor Alexander Schmidt (rqc908)
Helga Rykov Ibsen (mcv462)

August 31, 2023

1 Short description of the database

Our database takes its point of departure in real life and is inspired by a Danish company Easy Translate that offers translation services to all types of clients.

In the diagram (Figure 1 below), there are two stakeholders: Translators and Clients. The database is therefore thought of as a common platform serving both — translators and clients.

In the Table `Translators`, the field `Education` determines a translator's translation fees so that: (1) Secondary School goes hand in hand with `Category 3` in `Category` table that defines the lowest fees, (2) BA of Humanities corresponds to `Category 2` in `Category` table that defines middle fees, and (3) MA of Humanities and PhD correspond to `Category 1` in `Category` table that defines the highest fees.

A Translator has her employments prior to working at Easy Translate - hence the two tables `Translator` and `TranslatorEmployment` are related to each their as one-to-many: one translator has many employments with the name of the company, position at that company, date of employment and date of dismissal.

The idea of having this table goes to the wish for the client to be able to view all the translators of the company from most to least experienced, and then make a choice of the right translator for her. This functionality has been implemented as the method `GetTranslatorCompetenceViaDapper()`: Retrieve List of German Translators and their Total Experience in Years. (cf. DISProject/EasyTranslate/Data/Services/TranslatorEmploymentService.cs)

The table `Translator_Competence` is a bit of an octopus table, because it unites three other tables and establishes many-to-many relationship between them. It references the table of Translators, the table of Languages and the table of Categories.

To be more specific, a translator may translates three different languages, which, in turn, may correspond to different levels of education of the given translator and hence reference three different fee categories in the `Category` table (cf. the code in the appendices A and B).

The table `Language` speaks for itself. And the table of `Category` contains the information about fees according to a specific category. There are only three of them, so this table actually contains only three rows (1st highest, 2nd middle and 3rd lowest). In accord with a translator's category, she would get her oral fee, written fee, etc.

Clients in the table of `Client` represent state institutions (ministries, municipalities, Danish Royal Police, Immigration Service) and private persons. The field `FeeMultiplier` is reserved for generous clients: that is, those clients that usually pay more than normal fees of the agency.

A client can create a task request and there is a one-to-many relation between `Client` and `Task`, as one client can have many tasks and one task has only one client. This functional dependency is implemented with "ON DELETE CASCADE" constraint for the foreign key ClientId in the `Task_Review` table. So, in this respect, `Task` is a weak entity.

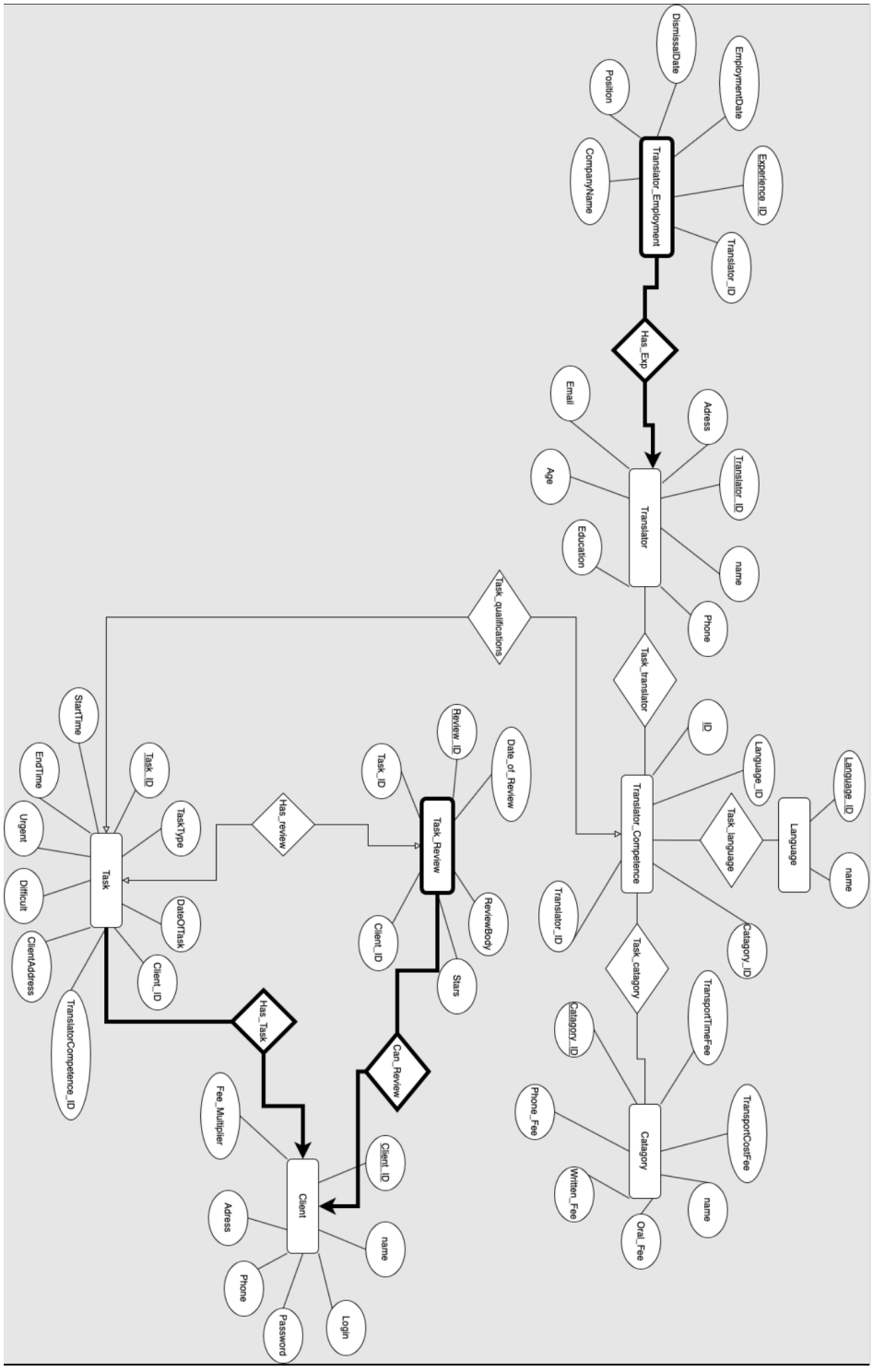

Figure 1: The E/R Diagram of Easy Translate: https://llk.dk/6twwlm

Moreover, a client can make a review of her tasks. There is a similar one-to-many relationship between `Task_Review` table and `Client` table, and `Task_Review` is a weak entity in relation to `Client`.

2 Considerations on denormalization of the database

The Easy Translate database presented in Figure 1 has been designed in accord with BCNF, meaning that it does not contain any relational redundancies and all entities are connected with each other through foreign key constraints between child tables and parent tables.

Yet, when writing our queries we have encountered a hurdle of 4-term JOINs that can be computationally expensive. In other words, in relational world, the database in BCNF is ideal, but in real life, it can greatly reduce the speed of read operations. For example, consider the following query: Get the average rating of all translators. In the original database in the diagram above, `Task_Review` table did not contain the TranslatorId field:

```
CREATE TABLE Task_Review (
    Id SERIAL PRIMARY KEY,
    DateOfReview DATE NOT NULL,
    Body TEXT,
    Stars INT DEFAULT 0 CHECK (Stars >= 1 AND Stars <= 5),
    TaskId INT REFERENCES Task(Id) ON DELETE CASCADE,
    ClientId INT REFERENCES Client(Id) ON DELETE NO ACTION
);
```

Thus, to calculate the average rating of all translators, whose tasks have been rated, we would have to go along the path: `Task_Review` —> `Task` —> `Translator_Competence`—> `Translator` and do a JOIN of four tables. Lets say we have got 2 million task reviews and 2000 translators, then this read-heavy operation might greatly impair our performance.

We have therefore opted for a compromise by adding some redundancy to our database, which we believe is closer to a real-life scenario, and decided to denormalize several relations: namely, we have added additional TranslatorId and LanguageId fields to `Task` and `Task_Review` tables. As already said, that has been done solely for decreasing the cost of such queries as (cf. the list of implemented queries in section 3.1):

1. Get translators with average ratings.
2. Get from most to least experienced translators.
3. Get from least to most expensive translators.

In order to add the additional fields to the already filled-in tables of `Task` and `Task_Review`, we made four `COALESCE` functions (cf. appendix C).

3 Web application

3.0.1 Considerations on the implementation of the database

As has been mentioned in section 1, the database has been thought of as serving two types of stakeholders: clients and translators. So, the idea goes to creating a web application for translators, on the one hand, and clients, on the other (not taking into account the side of the agency administration itself that must necessarily also have access to the web site in order to administer new clients, new translators, etc.).

After having logged in, a translator would access her Easy Translate site, while a client would be sent to a different Easy Translate site for clients. They would both have access to a bunch of functionalities there. Consider, for example, different user stories that have been designed separately for clients and translators.
A list of possible client actions:

1. Register a new client.
2. **Create a task request.**
3. Log in.
4. **View my (existing if any) tasks.**
5. **Post a task review.**
6. View all English translators, their task types and their task fees.
7. View all Russian and Ukrainian translators, their task types and their task fees.
8. **View from most to least experienced Spanish and German translators.**
9. View from most to least experienced all English translators who take the lowest fee.
10. View all most experienced translators who received bad task reviews.
11. View all written task reviews that have received less than 3 stars.
12. View all oral task reviews that have received 5 stars from 01.05.2022 to 01.05.2022.
13. **View all German translators who received rating $>$ 3.**
14. View all translators who performed phone translation tasks in Ukrainian in the period from 01.05.2020 to 01.05.2023 and received more than 3 stars.
15. View all translators who performed oral tasks for courts in English and received positive reviews.
16. View all Russian, Spanish and Arabic translators who performed written tasks for courts and police and received top reviews.
17. View translators who performed all sort of tasks for private clients in Swedish and received good reviews.
18. Read their task reviews.
19. View all English translators who received top reviews for all tasks and take the highest fee.

20. View Korean, Thai and Vietnamese translators who received top task reviews and take the lowest fee.
21. View translators and languages used that received bad task reviews.
22. View those translators' fees.
23. ...

A list of possible translator actions:

1. Log in into Easy Translate website.
2. View my date of employment at Easy Translate, languages and fees.
3. View all employments (companies) including the present one and calculate the overall time of my career as a translator.
4. Calculate the average time of employment at each company.
5. View my all performed tasks and respective clients, but not fees.
6. View salary earned in 2022.
7. View the year with least money earned.
8. View the period with most money earned at Easy Translate.
9. View clients who posted best reviews to my tasks.
10. View clients and their reviews who posted bad reviews to my tasks.
11. View the client who paid me most at Easy Translate.
12. Find the company that paid me most.
13. ...

And yet, to develop such a web application would be immensely time consuming, since a dynamic and editable implementation of our database requires months and months of work. Due to time constrains of the project at hand, we have opted for implementing the client side of the database alone. Some of the functionalities have been implemented as static. For example, we cannot add new languages to the existing table of `Languages`, but a client can view a table of all the languages of Easy Translate. Other functionalities have been implemented dynamically (cf section 3.1 below), allowing a client to add a new task or task review.

It's been decided that all interaction with the project's web site happens after an already existing client has logged in, because we found out that the implementation of the create-new-client and log-in functionalities would take us outside of the confines of the given project. Therefore, we have chosen an arbitrary client with the following fields, necessary for creating new tasks and posting task reviews:

Id	Name
22	Home Travel Agency

3.1 Implemented queries

For the project at hand, the following queries have been implemented:

1. Get all translators, language and category.
2. Get English, Russian and German translators.
3. Get all available languages.
4. View all existing client tasks.
5. Add a task.
   Note: combination of TranslatorId and LanguageId should exist in `Translator_Competence` table!
6. Add a task review.
7. Get translator fees.
8. Get the average rating of English, German and Russian translators.
9. Get German translators and their total experience in years. Order the results from most to least experienced.
10. Get German translators with the lowest fees.
11. Get German translators' fees. Order the results from the lowest to the highest fees.

The queries above have been implemented in Data project, which is part of the Easy Translate project and the actual implementations can be found in the folder `Services` (cf. /DISProject/EasyTranslate/Data/Services).

3.2 Easy Translate: A Multilayered Translation Application in C#

We have implemented Easy Translate as a comprehensive solution implemented in C#, comprising of several interlinked projects, each playing a distinct role. The overall system is structured using a multi-tier architecture, allowing for a clean separation of concerns, improved maintainability, and a streamlined development workflow.

1. Solution File:

The entire system is contained within a single solution file, linking all the sub-projects together. This provides a convenient way to manage the inter-project dependencies, streamline the build process, and facilitate unified debugging across all the projects.

2. Console Project:

The Console Project serves as a testing ground for various services provided by the Data Project. It consumes the services exposed by the Data Project to validate their functionality and ensure that they perform as expected. This project is crucial for our continuous testing process, assisting in identifying bugs and ensuring the integrity of the system throughout the development phase.

3. Data Project:

The Data Project acts as the backbone of the Easy Translate application, encompassing various classes and services that constitute the business logic layer. It encapsulates the core functionality of the translation process, housing methods with encapsulated queries in SQL and data processing routines necessary for the translation services. By separating these concerns into a dedicated project, we are able to improve reusability and make the system more modular.

4. Blazor Web Application Project:

The Blazor Web Application Project provides the frontend for the Easy Translate site, delivering an interactive and user-friendly interface for users to access the translation services. Leveraging the capabilities of Blazor, the application is a single-page app (SPA) that allows real-time interaction with the users. This project references the Data Project to provide the frontend with necessary data and functionalities.

By structuring our solution in this manner, we have a streamlined development process, clear separation of concerns, and an organized, manageable codebase. This architecture fosters a development environment that is conducive to testing, maintenance, and potential future expansion of the Easy Translate system.

4 How to run the project

We have concocted scripts for creating and inserting into the database **easy_translate**. Besides this, to run the project there are some dependencies.

4.1 Dependencies

The following needs to be instantiated and initialized, for the project to run as intended:

Blazor server / project

1. .NET, preferably .NET 7.203 as the solution was built on this version.
2. Tool for running .NET solutions (any terminal, Visual Studio, Rider, etc.).

Database - PostgreSQL

4.2 Creating and populating the database

4.2.1 Using our script

Prerequisites to run the `Windows_DB_Script.cmd` **or** `Bash_DB_Script.sh`:

1. The user postgres exists, *with exactly* password `1234`

The scripts drops if exists, creates and populates the database easy_translate. To run the script, either double click the script files corresponding your operating system (`.cmd` file on Windows, `.sh` on Mac or Linux), or run the following commands

`.\Windows_DB_Script.cmd` on Windows,

1. `chmod +x Bash_DB_Script.sh` (necessary on Mac)
2. `.\Bash_DB_Script.sh` on Mac / Linux

in the terminal from the `DISProject` folder.

4.2.2 Manually inserting the SQL

If for some reason the above can not be tried, the SQL file `/PostgreSQL/Easy_Translate.sql` contains the SQL which creates the tables and the necessary functions and seeds the database with data (inserts the static data which we have generated for the purposes of this project). Here follows two examples of how to insert the data into the database using:

PSQL (Shell): `\i <preceding path here>/DISProject/PostgreSQL/Easy_Translate.sql`

Terminal: `psql -U <your postgres user here> -f <preceding path here>/DISProject/PostgreSQL/Easy_Translate.sql`

4.3 Running the solution

NOTE: Postgres port The **default port is set to 5433. If this is not your default postgres port, you can either change your postgres port or the port in the Blazor project** (the database connection lies in the `/EasyTranslate/WebProject/Program.cs` if you wish to change it yourself).

To run the Blazor Project in an IDE, select the Easy_Translate solution, then select "BlazorProject" from the list of executable projects and press the "run" button.

Another option is to navigate to the `WebProject` folder and run the command `dotnet run`. This should build and run the project. To view the front-end, navigate the given link in the console: https://localhost:7177/

The front-end only cooperates fully, if connected to the populated (only tables are required) PostgreSQL database.

4.4 Interface differences

The front end looks different, depending on your operating system - at least this is our experience. We assume this has something to do with what operating system the project was created on, as newly created projects on Windows looks like the template. Just keep in mind that this was not intended, but some unexpected side-effect of using Blazor (which was undocumented and unprecedented, when looking for potential fixes for this on the internet). The functionality remains consistent.

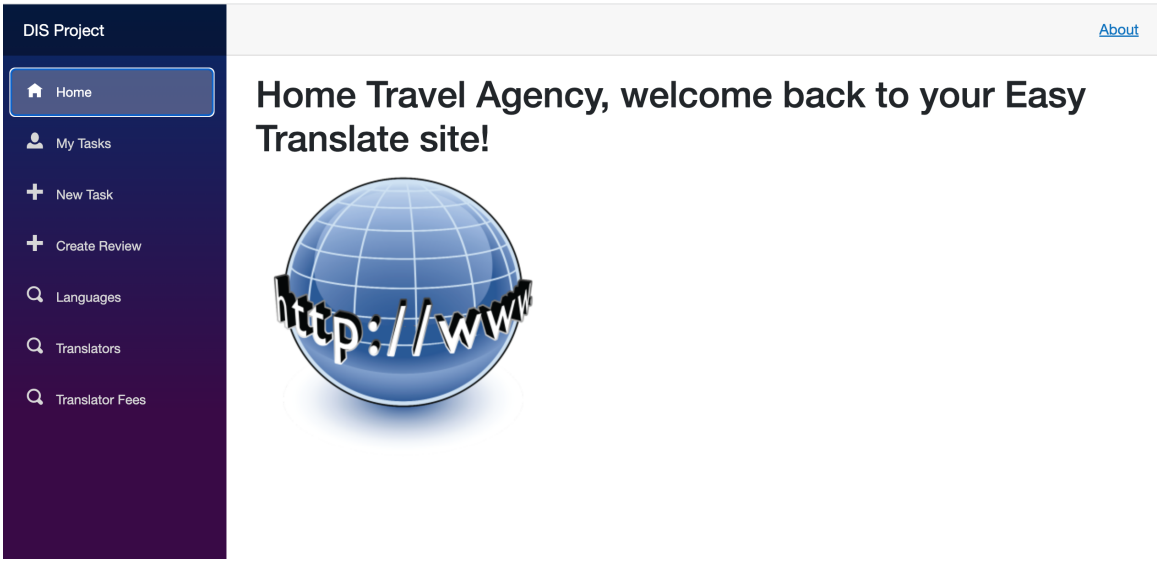

Figure 2: The user interface layout of the front end on the Mac operating system

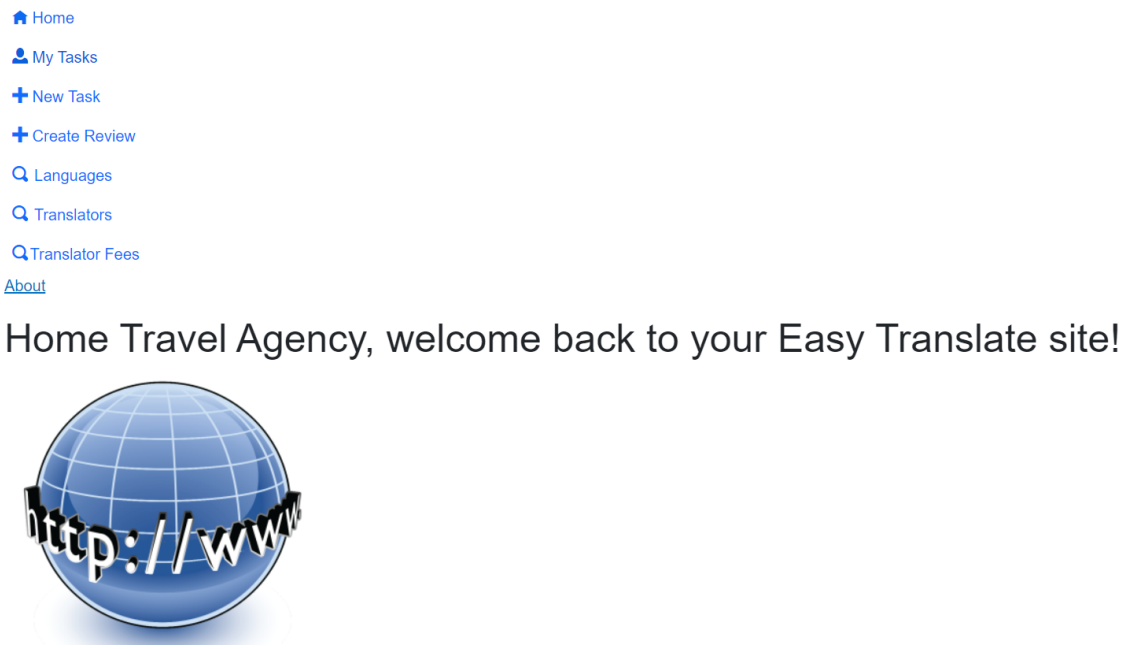

Figure 3: The user interface layout of the front end on the Windows operating system

A Appendix: SQL — Create Tables

```
CREATE TABLE Client (
    Id SERIAL PRIMARY KEY,
    ContactName VARCHAR(30),
    Login TEXT NOT NULL,
    Password TEXT NOT NULL,
    Tlf VARCHAR(20) UNIQUE,
    CityAddress VARCHAR(30),
    Street VARCHAR(50),
    HouseNr VARCHAR(20),
    FeeMultiplier FLOAT
);

CREATE TABLE Category (
    Id SERIAL PRIMARY KEY,
    CategoryName VARCHAR(11) NOT NULL,
    OralFee FLOAT,
    WrittenFee FLOAT,
    PhoneFee FLOAT,
    TransportCostFee FLOAT,
    TransportTimeFee FLOAT
);

CREATE TABLE Translator (
    Id SERIAL PRIMARY KEY,
    FirstName VARCHAR(30) NOT NULL,
    LastName VARCHAR(30) NOT NULL,
    Age INT DEFAULT 18 CHECK (Age >= 18 AND Age <= 70),
    CityAddress VARCHAR(30),
    Street VARCHAR(50),
    HouseNr VARCHAR(20),
    Email VARCHAR(50) UNIQUE,
    Tlf VARCHAR(20) UNIQUE,
    Education TEXT NOT NULL
);

CREATE TABLE Language (
    Id SERIAL PRIMARY KEY,
    NameOfLang VARCHAR(20) NOT NULL
);

CREATE TABLE Translator_Competence (
    Id SERIAL PRIMARY KEY,
    TranslatorId INT REFERENCES Translator(Id),
    LanguageId INT REFERENCES Language(Id),
    CategoryId INT REFERENCES Category(Id)
);
```

```
CREATE TABLE Task (
    Id SERIAL PRIMARY KEY,
    TaskType VARCHAR(30) NOT NULL,
    DateOfTask DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    Urgent INT,
    Difficult INT,
    CityAddress VARCHAR(30) NULL,
    Street VARCHAR(50) NULL,
    HouseNr VARCHAR(20) NULL,
    TranslatorCompetenceID INT REFERENCES Translator_Competence(Id),
    ClientId INT REFERENCES Client(Id) ON DELETE CASCADE,
    TranslatorId INT REFERENCES Translator(Id) ON DELETE NO ACTION,
    LanguageID INT REFERENCES Language(Id) ON DELETE NO ACTION
);

CREATE TABLE Task_Review (
    Id SERIAL PRIMARY KEY,
    DateOfReview DATE NOT NULL,
    Body TEXT,
    Stars INT DEFAULT 0 CHECK (Stars >= 1 AND Stars <= 5),
    TaskId INT REFERENCES Task(Id) ON DELETE CASCADE,
    ClientId INT REFERENCES Client(Id) ON DELETE NO ACTION,
    TranslatorId INT REFERENCES Translator(Id) ON DELETE NO ACTION,
    LanguageID INT REFERENCES Language(Id) ON DELETE NO ACTION
);

CREATE TABLE Translator_Employment (
    Id SERIAL PRIMARY KEY,
    EmploymentDate DATE NOT NULL,
    DismissalDate DATE,
    Position TEXT NOT NULL,
    CompanyName TEXT NOT NULL,
    TranslatorID INT REFERENCES Translator(Id) ON DELETE CASCADE
);
```

B Appendix: SQL — Fill in Tables

```
INSERT INTO Category (Id, CategoryName, OralFee, WrittenFee, PhoneFee, TransportCostFe
VALUES (1, 'Category␣1', 410, 4.5, 400, 205, 3.13),
       (2, 'Category␣2', 290, 3.2, 250, 145, 3.13),
       (3, 'Category␣3', 110, 2, 100, 55, 3.13);
```

```
INSERT INTO Translator (FirstName, LastName, Age, CityAddress,
Street, HouseNr, Email, Tlf, Education)
VALUES
    ('Anni', 'Hansen', 25, 'Copenhagen', 'Vigerslev␣Alle', '123',
    'annihansen@gmail.com', '3012␣1843', 'Secondary␣School'),
    ('Jane', 'Smith', 30, 'Aarhus', 'Hilleroedgade', '15',
    'janesmith@gmail.com', '3012␣1844', 'BA␣of␣Humanities'),
    ('David', 'Andersen', 35, 'Odense', 'Fengersvej', '10',
    'davidandersen@gmail.com', '3012␣1845', 'MA␣of␣Humanities'),
    ('Lars', 'Hansen', 59, 'Aalborg', 'Scharlingsvej', '1',
    'larshansen@gmail.com', '3012␣1846', 'PhD'),
    ('Mette', 'Nielsen', 68, 'Esbjerg', 'Ramsingsvej', '202',
    'mettenielsen@gmail.com', '3012␣1847', 'Secondary␣School'),
    ('Anders', 'Pedersen', 42, 'Frederiksberg', 'Tove␣Maes␣Vej',
    '65', 'anderspedersen@gmail.com', '3012␣1848', 'BA␣of
␣␣␣␣Humanities'),
    ('Charlotte', 'Jensen', 39, 'Horsens', 'Lyshoej␣Alle', '60',
    'charlottejensen@gmail.com', '3012␣1849', 'MA␣of␣Humanities'),
    ('Erik', 'Mortensen', 45, 'Randers', 'Jagtvej', '196',
    'erikmortensen@gmail.com', '3012␣1850', 'PhD'),
        ('Ahmed', 'Khalid', 31, 'Copenhagen', 'Bøgegade', '601',
    'ahmedkhalid@gmail.com', '4512␣1851', 'Secondary␣School'),
    ('Fatima', 'Ali', 45, 'Aarhus', 'Elmegade', '702',
    'fatimaali@gmail.com', '3012␣1852', 'BA␣of␣Humanities'),
    ('Mohammed', 'Hassan', 56, 'Odense', 'Ahornvej', '803',
    'mohammedhassan@gmail.com', '3112␣1853', 'MA␣of␣Humanities'),
    ('Aisha', 'Abdullah', 29, 'Esbjerg', 'Lindevej', '904',
    'aishaabdullah@gmail.com', '3012␣1854', 'PhD'),
    ('Youssef', 'Saeed', 36, 'Copenhagen', 'Egevej', '1005',
    'youssefsaeed@gmail.com', '3312␣1855', 'Secondary␣School'),
        ('Ivan', 'Sokolov', 33, 'Copenhagen', 'Bøgevej', '601',
 'ivansokolov@gmail.com', '3012␣1856', 'Secondary␣School'),
    ('Elena', 'Popova', 28, 'Aarhus', 'Elmevej', '702',
    'elenapopova@gmail.com', '3812␣1857', 'BA␣of␣Humanities'),
    ('Sergei', 'Ivanov', 36, 'Odense', 'Ahornvej', '803',
    'sergeiivanov@gmail.com', '3912␣1858', 'MA␣of␣Humanities'),
    ('Natalia', 'Smirnova', 31, 'Esbjerg', 'Lindetræsvej', '904',
    'nataliasmirnova@gmail.com', '4012␣1859', 'PhD'),
    ('Dmitri', 'Kuznetsov', 67, 'Copenhagen', 'Egevej', '1005',
```

```
    'dmitrikuznetsov@gmail.com', '6012 1860', 'Secondary School'),
        ('Helga', 'Ibsen', 41, 'Copenhagen', 'Lundtoftegade', '83',
 'helgaibsen@gmail.com', '6033 1860', 'BA of Humanities'),
        ('Senta', 'Abrahamsen', 65, 'Hellerup', 'Mosevej', '102',
 'sentaabrahamsen@gmail.com', '3412 2260', 'Secondary School')
;

INSERT INTO Language (Id, NameOfLang)
VALUES (1, 'English'),
       (2, 'Spanish'),
       (3, 'French'),
       (4, 'German'),
       (5, 'Mandarin'),
       (6, 'Arabic'),
       (7, 'Portuguese'),
       (8, 'Russian'),
       (9, 'Italian'),
       (10, 'Dutch'),
       (11, 'Swedish'),
       (12, 'Norwegian'),
       (13, 'Finnish'),
       (14, 'Danish'),
       (15, 'Greek'),
       (16, 'Turkish'),
       (17, 'Japanese'),
       (18, 'Korean'),
       (19, 'Thai'),
       (20, 'Vietnamese'),
       (21, 'Hindi'),
       (22, 'Bengali'),
       (23, 'Urdu'),
       (24, 'Punjabi'),
       (25, 'Persian'),
       (26, 'Indonesian'),
       (27, 'Malay'),
       (28, 'Filipino'),
       (29, 'Swahili'),
       (30, 'Czech'),
       (31, 'Polish');


INSERT INTO Translator_Competence (Id, TranslatorId, LanguageId, CategoryId)
VALUES
    (1, 1, 1, 1),
    (2, 1, 2, 1),
    (3, 1, 3, 1),
    (4, 2, 15, 2),
    (5, 2, 16, 2),
    (6, 2, 31, 2),
```

```
(7, 3, 4, 3),
(8, 3, 2, 2),
(9, 3, 6, 3),
(10, 4, 10, 1),
(11, 4, 11, 3),
(12, 4, 12, 2),
(13, 5, 13, 1),
(14, 5, 8, 2),
(15, 5, 15, 1),
(16, 6, 16, 3),
(17, 6, 17, 2),
(18, 6, 4, 3),
(19, 7, 9, 1),
(20, 7, 7, 1),
(21, 7, 5, 1),
(22, 8, 8, 2),
(23, 8, 19, 2),
(24, 8, 4, 3),
(25, 9, 20, 3),
(26, 9, 21, 3),
(27, 9, 30, 3),
(28, 10, 8, 1),
(29, 10, 4, 3),
(30, 10, 22, 3),
(31, 11, 23, 2),
(32, 11, 4, 2),
(33, 11, 25, 2),
(34, 12, 26, 3),
(35, 12, 2, 3),
(36, 12, 3, 3),
(37, 13, 26, 1),
(38, 13, 7, 1),
(39, 13, 9, 1),
(40, 14, 8, 2),
(41, 14, 4, 2),
(42, 14, 2, 2),
(43, 15, 18, 3),
(44, 15, 1, 3),
(45, 15, 5, 3),
(46, 16, 1, 1),
(47, 16, 4, 1),
(48, 16, 8, 3),
(49, 17, 4, 2),
(50, 17, 13, 2),
(51, 17, 8, 3),
(52, 18, 4, 3),
(53, 18, 2, 3),
(54, 18, 3, 3),
(55, 19, 1, 1),
```

```
    (56, 19, 3, 1),
    (57, 19, 8, 3),
    (58, 20, 4, 2),
    (59, 20, 2, 2),
    (60, 20, 3, 3);


-- Insert courts
INSERT INTO Client (Id, ContactName, Login, Password, Tlf, CityAddress,
Street, HouseNr, FeeMultiplier)
VALUES
    (1, 'Copenhagen City Court', 'cph_court', 'court123', '20 40 60 80',
        'Copenhagen', 'Hovedgaden', '123', 1.2),
    (2, 'Hilleroed Court', 'hilleroed_court', 'court456', '21 41 61 81',
        'Hilleroed', 'Egevej', '456', 1.0),
    (3, 'Helsingoer Court', 'helsingoer_court', 'court789', '22 42 62 82',
        'Helsingoer', 'Bøgevej', '789', 1.1),
    (4, 'Roskilde Court', 'roskilde_court', 'courtabc', '23 43 63 83',
        'Roskilde', 'Ahornvej', '101', 0.9),
    (5, 'Landsbyret', 'landsbyret', 'courttuv', '24 44 64 84',
        'Copenhagen', 'Borgergade', '1', 1.5),
    (6, 'Herning Court', 'herning_court', 'courtxyz', '25 45 65 85',
        'Herning', 'Elmegade', '303', 1.1),
    (7, 'Viborg Court', 'viborg_court', 'court123abc', '26 46 66 86',
        'Viborg', 'Birkelunden', '404', 1.0),
    (8, 'Aalborg Court', 'aalborg_court', 'court456def', '27 47 67 87',
        'Aalborg', 'Elmevej', '505', 0.9),
    (9, 'Naestved Court', 'naestved_court', 'court789ghi', '28 48 68 88',
        'Naestved', 'Kastanievej', '601', 1.2),
    (10, 'Sonderborg Court', 'sonderborg_court', 'courtabcjkl', '29 49 69 89',
        'Sonderborg', 'Lindegade', '702', 1.3);

-- Insert police stations
INSERT INTO Client (Id, ContactName, Login, Password, Tlf, CityAddress, Street,
HouseNr, FeeMultiplier)
VALUES
    (11, 'Copenhagen Police Station', 'cph_police',
        'password123', '24 56 78 90', 'Copenhagen', 'Hovedgaden', '123', 0.0),
    (12, 'Aarhus Police Station', 'aarhus_police',
        'password456', '28 36 54 72', 'Aarhus', 'Egevej', '456', 0.0),
    (13, 'Odense Police Station', 'odense_police',
        'password789', '23 45 67 89', 'Odense', 'Ahornvej', '789', 0.0),
    (14, 'Aalborg Police Station', 'aalborg_police',
        'passwordabc', '25 35 47 59', 'Aalborg', 'Elmevej', '101', 0.0),
    (15, 'Esbjerg Police Station', 'esbjerg_police',
        'passworddef', '27 38 49 50', 'Esbjerg', 'Lindevej', '202', 0.0),
    (16, 'Frederiksberg Police Station', 'frederiksberg_police',
        'pass456word', '22 33 44 55', 'Frederiksberg', 'Elmevej', '303', 0.0),
    (17, 'Horsens Police Station', 'horsens_police',
```

```
        'pass789word', '26 37 48 59', 'Horsens', 'Fyrstegade', '404', 0.1),
    (18, 'Randers Police Station', 'randers_police',
        'passabcword', '29 39 49 59', 'Randers', 'Grovevej', '505', 0.0),
    (19, 'Helsingoer Station', 'cph_central_police',
        'passdefword', '21 31 41 51', 'Helsingoer', 'Bøgegade', '10', 0.0),
    (20, 'Aarhus Central Police Station', 'aarhus_central_police',
        'passxyzword', '23 33 43 53', 'Aarhus', 'Elmegade', '702', 0.0);


-- Insert Immigration Service
INSERT INTO Client (Id, ContactName, Login, Password, Tlf, CityAddress, Street,
HouseNr, FeeMultiplier)
VALUES
    (21, 'Immigration Service', 'immigration_service', 'immigration123',
        '30 50 70 90', 'Naestved', 'Hovedgaden', '789', 0.0);

-- Insert Home Travel Agency
INSERT INTO Client (Id, ContactName, Login, Password, Tlf, CityAddress, Street,
HouseNr, FeeMultiplier)
VALUES
    (22, 'Home Travel Agency', 'home_travel', 'travel456',
        '31 51 71 91', 'Copenhagen', 'Strandvejen', '123', 0.0);

-- Insert private persons
INSERT INTO Client (Id, ContactName, Login, Password, Tlf, CityAddress, Street,
HouseNr, FeeMultiplier)
VALUES
    (23, 'Sophia Larsen', 'sophial', 'password135', '47 57 67 77',
        'Copenhagen', 'Rosenvej', '67', 0.0),
    (24, 'Frederik Nielsen', 'frederikn', 'password246', '48 58 68 78',
        'Aarhus', 'Ahornvej', '89', 0.0),
    (25, 'Emma Christensen', 'emmac', 'password357', '49 59 69 79',
        'Odense', 'Birkelunden', '10', 0.0),
    (26, 'Lucas Jensen', 'lucasj', 'password468', '50 60 70 80',
        'Esbjerg', 'Elmegade', '32', 0.0),
    (27, 'Maja Pedersen', 'majap', 'password321', '43 53 63 73',
        'Aalborg', 'Fyrvej', '78', 0.0),
    (28, 'Nikolaj Rasmussen', 'nikolajr', 'password654', '44 54 64 74',
        'Esbjerg', 'Bøgevej', '90', 1.0),
    (29, 'Laura Andersen', 'lauraa', 'password987', '45 55 65 75',
        'Horsens', 'Lærkevej', '23', 0.0),
    (30, 'Oliver Madsen', 'oliverm', 'password246', '46 56 66 76',
        'Randers', 'Kastanievej', '45', 0.9),
    (31, 'Lars Jensen', 'larsj', 'password123', '40 50 60 70',
        'Copenhagen', 'Elmegade', '12', 0.0),
    (32, 'Emma Nielsen', 'emman', 'password456', '41 51 61 71',
        'Aarhus', 'Birkevej', '34', 0.9),
    (33, 'Sofie Hansen', 'sofieh', 'password789', '42 52 62 72',
        'Odense', 'Ahornvej', '56', 0.0);
```

```
INSERT INTO Task (Id, TaskType, DateOfTask, StartTime, EndTime, Urgent,
 Difficult, CityAddress, Street, HouseNr, TranslatorCompetenceID, ClientId,
 TranslatorId, LanguageID)
VALUES
    (1, 'Oral Interpretation', '2020-05-10', '10:00', '12:00', 0, 0,
        'Copenhagen', 'Hovedgaden', '123', 1, 10, (SELECT transfun(1)),
        (SELECT langfun(1))),
    (2, 'Oral Interpretation', '2022-09-11', '14:00', '16:00', 1, 1,
        'Hilleroed', 'Egevej', '456', 4, 2, (SELECT transfun(4)),
        (SELECT langfun(4))),
    (3, 'Oral Interpretation', '2020-12-12', '09:00', '11:00', 0, 1,
        'Helsingoer', 'Bøgevej', '789', 7, 3, (SELECT transfun(7)),
        (SELECT langfun(7))),
    (4, 'Phone Translation', '2022-11-01', '17:00', '19:00', 0, 1,
        'Roskilde', 'Ahornvej', '101', 14, 4, (SELECT transfun(14)),
        (SELECT langfun(14))),
    (5, 'Oral Interpretation', '2021-01-14', '14:00', '15:30', 1, 1,
        'Copenhagen', 'Borgergade', '1', 3, 5, (SELECT transfun(3)),
        (SELECT langfun(3))),
    (6, 'Written Translation', '2021-02-15', '06:00', '23:00', 0, 0,
        'Viborg', 'Birkelunden', '404', 18, 7, (SELECT transfun(18)),
        (SELECT langfun(18))),
    (7, 'Oral Interpretation', '2021-03-02', '09:00', '17:00', 0, 1,
        'Aalborg', 'Elmevej', '505', 22, 8, (SELECT transfun(22)),
        (SELECT langfun(22))),
    (8, 'Oral Interpretation', '2021-05-17', '14:00', '18:00', 1, 1,
        'Aarhus', 'Egevej', '456', 24, 10, (SELECT transfun(24)),
        (SELECT langfun(24))),
    (9, 'Phone Translation', '2022-06-18', '10:00', '12:00', 0, 0,
        'Esbjerg', 'Lindevej', '202', 28, 12, (SELECT transfun(28)),
        (SELECT langfun(28))),
    (10, 'Phone Translation', '2020-06-19', '13:00', '15:00', 1, 0,
        'Randers', 'Grovevej', '505', 29, 14, (SELECT transfun(29)),
        (SELECT langfun(29))),
    (11, 'Phone Translation', '2021-07-20', '16:00', '18:00', 0, 0,
        'Horsens', 'Fyrstegade', '404', 32, 1, (SELECT transfun(32)),
        (SELECT langfun(32))),
    (12, 'Written Translation', '2020-01-21', '07:00', '21:00', 0, 1,
        'Copenhagen', 'Rosenvej', '67', 40, 20, (SELECT transfun(40)),
        (SELECT langfun(40))),
    (13, 'Written Translation', '2021-02-22', '07:00', '23:00', 1, 0,
        'Aarhus', 'Ahornvej', '89', 41, 21, (SELECT transfun(41)),
        (SELECT langfun(41))),
    (14, 'Written Translation', '2022-11-23', '11:00', '13:00', 0, 1,
        'Odense', 'Birkelunden', '10', 44, 22, (SELECT transfun(44)),
        (SELECT langfun(44))),
    (15, 'Phone Translation', '2022-12-26', '08:00', '10:00', 0, 0,
```

```
    'Esbjerg', 'Elmegade', '32', 9, 23, (SELECT transfun(9)),
    (SELECT langfun(9))),
(16, 'Oral Interpretation', '2021-07-25', '12:00', '18:00', 1, 1,
    'Aalborg', 'Fyrvej', '78', 46, 24, (SELECT transfun(46)),
    (SELECT langfun(46))),
(17, 'Oral Interpretation', '2023-04-11', '12:00', '18:00', 0, 0,
    'Esbjerg', 'Bøgevej', '90', 2, 25, (SELECT transfun(2)),
    (SELECT langfun(2))),
(18, 'Phone Translation', '2023-02-04', '11:00', '14:00', 0, 0,
    'Horsens', 'Lærkevej', '23', 47, 26, (SELECT transfun(47)),
    (SELECT langfun(47))),
(19, 'Written Translation', '2020-02-04', '06:00', '23:00', 1, 0,
    'Randers', 'Kastanievej', '45', 48, 27, (SELECT transfun(48)),
    (SELECT langfun(48))),
(20, 'Written Translation', '2021-08-15', '06:00', '23:00', 0, 0,
    'Copenhagen', 'Elmegade', '12', 49, 28, (SELECT transfun(49)),
    (SELECT langfun(49))),
(21, 'Phone Translation', '2023-01-12', '15:00', '19:00', 0, 0,
    'Aarhus', 'Birkevej', '34', 30, 29, (SELECT transfun(30)),
    (SELECT langfun(30))),
(22, 'Oral Interpretation', '2020-09-18', '10:00', '16:00', 0, 0,
    'Odense', 'Ahornvej', '56', 24, 30, (SELECT transfun(24)),
    (SELECT langfun(24))),
(23, 'Phone Translation', '2020-10-12', '10:00', '12:00', 0, 0,
    'Copenhagen', 'Strandvejen', '123', 29, 22, (SELECT transfun(29)),
    (SELECT langfun(29))),
(24, 'Oral Interpretation', '2021-02-12', '10:00', '15:00', 0, 1,
    'Copenhagen', 'Strandvejen', '123', 51, 22, (SELECT transfun(51)),
    (SELECT langfun(51))),
(25, 'Oral Interpretation', '2023-02-27', '09:00', '17:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 52, 21,
    (SELECT transfun(52)), (SELECT langfun(52))),
(26, 'Oral Interpretation', '2022-08-03', '09:00', '16:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 53, 21, (SELECT transfun(53)),
    (SELECT langfun(53))),
(27, 'Oral Interpretation', '2022-11-08', '09:00', '15:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 11, 21, (SELECT transfun(11)),
    (SELECT langfun(11))),
(28, 'Oral Interpretation', '2022-07-08', '09:00', '15:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 55, 21, (SELECT transfun(55)),
    (SELECT langfun(55))),
(29, 'Oral Interpretation', '2023-05-08', '09:00', '15:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 57, 21, (SELECT transfun(57)),
    (SELECT langfun(57))),
(30, 'Oral Interpretation', '2022-03-18', '09:00', '15:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 58, 21, (SELECT transfun(58)),
    (SELECT langfun(58))),
(31, 'Oral Interpretation', '2021-05-12', '09:00', '15:00', 0, 0,
    'Naestved', 'Hovedgaden', '789', 8, 21, (SELECT transfun(8)),
```

```
        (SELECT langfun(8))),
    (32, 'Written Translation', '2021-09-22', '06:00', '23:00', 1, 0,
        'Frederiksberg', 'Elmevej', '303', 15, 6, (SELECT transfun(15)),
        (SELECT langfun(15))),
    (33, 'Written Translation', '2022-06-20', '06:00', '23:00', 0, 0,
        'Naestved', 'Kastanievej', '601', 32, 9, (SELECT transfun(32)),
        (SELECT langfun(32))),
    (34, 'Written Translation', '2022-10-11', '06:00', '23:00', 0, 0,
        'Odense', 'Ahornvej', '789', 7, 13,
        (SELECT transfun(7)), (SELECT langfun(7))),
    (35, 'Written Translation', '2023-01-04', '06:00', '23:00', 1, 0,
        'Esbjerg', 'Lindevej', '202', 1, 15, (SELECT transfun(1)),
        (SELECT langfun(1)));


INSERT INTO Task_Review (Id, DateOfReview, Body, Stars, TaskId, ClientId, TranslatorId
VALUES
    (1, '2020-05-22', 'Fremragende arbejde!',
        5, 1, 1, (SELECT transfun1(1)), (SELECT langfun1(1))),
    (2, '2022-10-23', 'Fantastisk tolkning!',
        5, 2, 2, (SELECT transfun1(2)), (SELECT langfun1(2))),
    (3, '2021-03-24', 'Meget tilfredsstillende oversættelse.', 4, 6, 3,
        (SELECT transfun1(6)), (SELECT langfun1(6))),
    (4, '2021-01-10', 'God telefontolkning.',
        4, 4, 4, (SELECT transfun1(4)), (SELECT langfun1(4))),
    (5, '2021-01-26', 'Ikke tilfreds med tidsstyring.
        Opfylde opgaven for sent.',
        2, 5, 5, (SELECT transfun1(5)), (SELECT langfun1(5))),
    (6, '2021-02-27', 'Fantastisk arbejde! Meget imponeret.', 5, 3, 6,
        (SELECT transfun1(3)), (SELECT langfun1(3))),
    (7, '2021-03-15', 'Udmærket tolkning.',
        3, 7, 7, (SELECT transfun1(7)), (SELECT langfun1(7))),
    (8, '2021-05-29', 'Flot oversættelse.',
        4, 8, 8, (SELECT transfun1(8)), (SELECT langfun1(8))),
    (9, '2022-06-30', 'Meget god telefontolkning.',
        4, 9, 9, (SELECT transfun1(9)), (SELECT langfun1(9))),
    (10, '2020-06-22', 'Manglede færdiggørelsen af
        opgaven til tiden.', 2, 10, 10,
        (SELECT transfun1(10)), (SELECT langfun1(10))),
    (11, '2021-07-26', 'Udmærket arbejde.',
        3, 11, 11, (SELECT transfun1(11)), (SELECT langfun1(11))),
    (12, '2020-02-02', 'God oversættelse.',
        4, 12, 12, (SELECT transfun1(12)), (SELECT langfun1(12))),
    (13, '2021-03-03', 'Udmærket tolkning.',
        3, 13, 13, (SELECT transfun1(13)), (SELECT langfun1(13))),
    (14, '2022-12-04', 'Fremragende arbejde!
        Vældig tilfreds.', 5, 17, 14,
        (SELECT transfun1(17)), (SELECT langfun1(17))),
    (15, '2022-12-28', 'Fantastisk oversættelse!',
```

```
        5, 15, 15, (SELECT transfun1(15)), (SELECT langfun1(15))),
    (16, '2023-02-06', 'Tilfredsstillende telefontolkning.',
        4, 21, 16,
        (SELECT transfun1(21)), (SELECT langfun1(21))),
    (17, '2023-04-20', 'Manglede at levere opgaven til
        tiden og indeholdte flere fejl.',
        2, 14, 17, (SELECT transfun1(14)), (SELECT langfun1(14))),
    (18, '2023-02-08', 'Godt udført arbejde.
        Kan godt finde på at bestille samme tolk igen.',
        4, 18, 18, (SELECT transfun1(18)), (SELECT langfun1(18))),
    (19, '2020-02-09', 'Meget tilfreds med oversættelsen.
        Udført til tiden.', 4, 19, 19,
        (SELECT transfun1(19)), (SELECT langfun1(19))),
    (20, '2021-08-20', 'Udmærket tolkning.',
        3, 20, 20, (SELECT transfun1(20)), (SELECT langfun1(20))),
    (21, '2023-01-15', 'Fantastisk arbejde!
        Kan varmt anbefales. Dygtig tolk.', 5, 31, 31,
        (SELECT transfun1(31)), (SELECT langfun1(31))),
    (22, '2021-09-28', 'God oversættelse.', 4, 32, 32,
        (SELECT transfun1(32)), (SELECT langfun1(32))),
    (23, '2022-06-23', 'Ikke særlig god oversættelse:
        tolken har lavet flere fejl med juridiske termer.',
        1, 33, 33, (SELECT transfun1(33)), (SELECT langfun1(33))),
    (24, '2022-10-24', 'Meget tilfreds med oversættelsen.', 5, 34, 20,
        (SELECT transfun1(34)), (SELECT langfun1(34))),
    (25, '2023-01-25', 'Fantastisk oversættelse!', 5, 35, 12,
        (SELECT transfun1(35)), (SELECT langfun1(35)));


INSERT INTO Translator_Employment (Id, EmploymentDate, DismissalDate, Position, Compan
VALUES
    (1, '2002-03-15', '2004-06-30', 'Tolk og oversætter', 'Dansk Translations', 1),
    (2, '2005-07-01', '2011-01-01', 'Oversætter', 'Sprogcentret', 2),
    (3, '2010-02-15', '2014-06-15', 'Tolk og oversætter', 'Language Solutions', 3),
    (4, '1997-01-01', '2002-01-01', 'Tolk', 'Translatørhuset', 4),
    (5, '2001-06-10', '2007-06-10', 'Oversætter', 'Oversættelsesbureauet', 5),
    (6, '2005-01-01', '2008-01-01', 'Tolk og oversætter', 'Sprogbroen', 6),
    (7, '2000-09-01', '2005-09-01', 'Tolk', 'Global Language Services', 7),
    (8, '2001-01-01', '2008-01-01', 'Oversætter', 'Polyglot Translations', 8),
    (9, '2003-08-15', '2005-02-15', 'Tolk og oversætter', 'Linguistica', 9),
    (10, '1996-02-01', '1998-02-01', 'Tolk', 'Interlang', 10),
    (11, '2013-04-15', '2014-03-01', 'Oversætter', 'Interlang', 11),
    (12, '1996-01-01', '2007-12-01', 'Tolk og oversætter', 'ProLingo', 12),
    (13, '1998-05-10', '2001-05-01', 'Oversætter', 'TransWorld', 13),
    (14, '1997-01-01', '2005-01-01', 'Tolk og oversætter',
        'Globale Kommunikation', 14),
    (15, '1996-01-15', '2006-08-10', 'Tolk', 'TransNation', 15),
    (16, '2002-01-01', '2011-12-01', 'Oversætter', 'LingoMasters', 16),
    (17, '2005-09-01', '2006-08-01', 'Tolk og oversætter', 'EuroLingua', 17),
```

```
(18, '2007-02-01', '2010-01-01', 'Oversætter', 'Interpret Solutions', 18),
(19, '2010-06-15', '2012-01-01', 'Tolk', 'Language Link', 19),
(20, '1996-01-01', '1999-12-15', 'Tolk og oversætter', 'MultiLingual', 20),
(21, '2006-01-15', '2009-12-31', 'Tolk og oversætter', 'Sprogcentret', 1),
(22, '2011-02-01', '2013-06-01', 'Oversætter', 'Dansk Translations', 2),
(23, '2014-06-15', '2016-01-15', 'Tolk og oversætter', 'Translatørhuset', 3),
(24, '2003-01-01', '2005-01-01', 'Tolk', 'Language Solutions', 4),
(25, '2008-06-10', '2010-06-10', 'Oversætter', 'Sprogbroen', 5),
(26, '2009-01-01', '2011-01-01', 'Tolk og oversætter','Translatørhuset', 6),
(27, '2006-09-01', '2010-09-01', 'Tolk', 'Sprogbroen', 7),
(28, '2009-01-01', '2010-01-01', 'Oversætter',
    'Global Language Services', 8),
(29, '2005-08-15', '2008-05-15', 'Tolk og oversætter', 'LingoMasters', 9),
(30, '2001-02-01', '2009-01-01', 'Tolk', 'TransWorld', 10),
(31, '2014-04-15', '2015-04-01', 'Oversætter', 'Language Link', 11),
(32, '2008-01-01', '2011-12-01', 'Tolk og oversætter', 'TransWorld', 12),
(33, '2001-05-10', '2006-05-05', 'Oversætter', 'EuroLingua', 13),
(34, '2005-01-01', '2009-12-01', 'Tolk og oversætter', 'Sprogbroen', 14),
(35, '2006-08-15', '2011-08-10', 'Tolk', 'LingoMasters', 15),
(36, '2012-01-01', '2016-12-01', 'Oversætter', 'Language Link', 16),
(37, '2006-09-01', '2007-08-01', 'Tolk og oversætter', 'MultiLingual', 17),
(38, '2010-02-01', '2012-01-01', 'Oversætter', 'Language Solutions', 18),
(39, '2012-01-15', '2012-07-10', 'Tolk', 'TransWorld', 19),
(40, '2000-01-01', '2009-12-01', 'Tolk og oversætter',
    'Dansk Translations', 20),
(41, '2010-03-15', '2012-06-30', 'Tolk og oversætter',
    'Dansk Translations', 1),
(42, '2013-07-01', '2017-05-01', 'Oversætter', 'Sprogcentret', 2),
(43, '2017-02-15', '2019-02-15', 'Tolk og oversætter',
    'Language Solutions', 3),
(44, '2012-01-01', '2019-01-01', 'Tolk', 'Translatørhuset', 4),
(45, '2015-06-10', '2019-06-10', 'Oversætter', 'Sprogbroen', 5),
(46, '2015-01-01', '2019-01-01', 'Tolk og oversætter',
    'Translatørhuset', 6),
(47, '2015-09-01', '2020-09-01', 'Tolk', 'Sprogbroen', 7),
(48, '2016-01-01', '2021-01-01', 'Oversætter',
    'Global Language Services', 8),
(49, '2016-08-15', '2022-03-15', 'Tolk og oversætter', 'LingoMasters', 9),
(50, '2014-02-01', '2019-02-01', 'Tolk', 'TransWorld', 10),
(51, '2018-04-15', '2020-04-01', 'Oversætter', 'Language Link', 11),
(52, '2018-01-01', '2020-12-01', 'Tolk og oversætter', 'TransWorld', 12),
(53, '2013-05-10', '2022-05-05', 'Oversætter', 'EuroLingua', 13),
(54, '2017-01-01', '2020-12-01', 'Tolk og oversætter', 'Sprogbroen', 14),
(55, '2014-08-15', '2020-08-10', 'Tolk', 'LingoMasters', 15),
(56, '2019-01-01', '2022-12-01', 'Oversætter', 'Language Link', 16),
(57, '2009-09-01', '2022-08-01', 'Tolk og oversætter', 'MultiLingual', 17),
(58, '2014-02-01', '2021-01-01', 'Oversætter', 'Language Solutions', 18),
(59, '2016-07-15', '2020-07-14', 'Tolk', 'TransWorld', 19),
(60, '2015-01-01', '2020-02-25', 'Tolk og oversætter',
```

```
    'Dansk Translations', 20),
(61, '2021-03-15', NULL, 'Tolk og oversætter', 'Easy Translate', 1),
(62, '2021-07-01', NULL, 'Oversætter', 'Easy Translate', 2),
(63, '2020-02-15', NULL, 'Tolk og oversætter', 'Easy Translate', 3),
(64, '2020-01-01', NULL, 'Tolk', 'Easy Translate', 4),
(65, '2020-06-10', NULL, 'Oversætter', 'Easy Translate', 5),
(66, '2020-01-01', NULL, 'Tolk og oversætter', 'Easy Translate', 6),
(67, '2021-09-01', NULL, 'Tolk', 'Easy Translate', 7),
(68, '2022-01-01', NULL, 'Oversætter', 'Easy Translate', 8),
(69, '2023-03-15', NULL, 'Tolk og oversætter', 'Easy Translate', 9),
(70, '2020-02-01', NULL, 'Tolk', 'Easy Translate', 10),
(71, '2020-04-15', NULL, 'Oversætter', 'Easy Translate', 11),
(72, '2021-01-01', NULL, 'Tolk og oversætter', 'Easy Translate', 12),
(73, '2022-05-10', NULL, 'Oversætter', 'Easy Translate', 13),
(74, '2021-01-01', NULL, 'Tolk og oversætter', 'Easy Translate', 14),
(75, '2020-08-15', NULL, 'Tolk', 'Easy Translate', 15),
(76, '2023-01-01', NULL, 'Oversætter', 'Easy Translate', 16),
(77, '2022-09-01', NULL, 'Tolk og oversætter', 'Easy Translate', 17),
(78, '2021-02-01', NULL, 'Oversætter', 'Easy Translate', 18),
(79, '2020-07-15', NULL, 'Tolk', 'Easy Translate', 19),
(80, '2020-03-01', NULL, 'Tolk og oversætter', 'Easy Translate', 20);
```

C Appendix: SQL — Functions for after-fill-in

```
-- Function for Task Table to get TranslatorId
CREATE FUNCTION transfun(p_id integer) RETURNS integer AS $$
    SELECT COALESCE(TranslatorId, 0)
    FROM Translator_Competence AS tc
    WHERE tc.id = p_id;
$$ LANGUAGE SQL;

-- Function for Task Table to get LanguageId
CREATE FUNCTION langfun(p_id integer) RETURNS integer AS $$
    SELECT COALESCE(LanguageId, 0)
    FROM Translator_Competence AS tc
    WHERE tc.id = p_id;
$$ LANGUAGE SQL;

-- Fun for Task_Review Table to get TranslatorId
CREATE OR REPLACE FUNCTION transfun1(p_id integer)
RETURNS integer AS $$
    SELECT COALESCE (TC.TranslatorId, 0)
    FROM Task AS T
    INNER JOIN Translator_Competence AS TC ON T.TranslatorCompetenceID = TC.Id
    WHERE T.Id = p_id;
$$ LANGUAGE SQL;

-- Fun for Task_Review Table to get LanguageId
CREATE OR REPLACE FUNCTION langfun1(p_id integer)
RETURNS integer AS $$
    SELECT COALESCE (TC.LanguageId, 0)
    FROM Task AS T
    INNER JOIN Translator_Competence AS TC ON T.TranslatorCompetenceID = TC.Id
    WHERE T.Id = p_id;
$$ LANGUAGE SQL;
```