Hurtownie danych – Projekt – Etap 02

PWr. Data: 25-26.04.2022

Student	254534@student.pwr.edu.pl	Ocena
Indeks	254534	
Imię	Kamila	
Nazwisko	Sproska	

Zestaw składa się z 2 zadań. Pamiętaj o podaniu nr. indeksu oraz imienia i nazwiska.

I. Wstępna specyfikacja wybranego tematu projektu

1. Tytuł projektu

Analiza zatrudnień programistów w Europie w latach 2018-2021

2. Charakterystyka dziedziny problemowej

Na zatrudnienie i stawki programistów może wpływać wiele przeróżnych czynników – od czynników społecznych, przez kraj zamieszkania, po czynniki na które pracownik ma wpływ takie jak doświadczenie czy edukacja. Są różne zapotrzebowania na umiejętności w różnych technologiach i na różne stanowiska, trudne jest z tego względu określenie w jakim kierunku najlepiej się dokształcać i jakich zarobków można się spodziewać w zależności od kwalifikacji pracownika.

2.1 Opis obszaru analizy (wybrany fragment dziedziny, przeznaczony do szczegółowej analizy i opracowania hurtowni danych)

Będziemy zajmować się szczegółową analizą zbioru danych dotyczących zarobków programistów w latach 2018-2021. Szczególną trzeba zwrócić uwagę na różnice w zarobkach względem kraju zatrudnienia, wykształcenia, lat doświadczenia, płci i ile i jakie programista zna języki obce, w jakim języku programowania pisze oraz jakiej jest specjalizacji.

2.2 Problemy

P1 – Brak świadomości programistów dotyczących popytu na pracowników o danych kwalifikacjach
P2 – Brak informacji dotyczącej wpływu czynników takich jak płeć i wiek na zarobki
P3 – Brak możliwości wybrania najlepszej ścieżki edukacji w celu polepszenia zarobków
P4 – Brak możliwości domagania się stawki odpowiadającej podejmowanemu stanowisku

2.3 Cel przedsięwzięcia

Określenie czynników wpływających na zarobki informatyków

2.3.1 Oczekiwania i potrzeby w zakresie wsparcia podejmowania decyzji (pytania badawcze)

Poprzez analizę danych chcemy uzyskać odpowiedzi na między innymi następujące pytania:

- Jak płeć i wiek wpływają na zarobki? Czy widoczne są wyraźne różnice?
- Programiści o jakiej specjalizacji zarabiają najwięcej?
- Dla jakiego rodzaju firmy zarobki są największe?
- Dla jakich krajów w których firmy prowadzą działalność najbardziej opłaca się pracować?
- Czy osoby o wyższym wykształceniu zarabiają więcej?
- Firmy jakiego rodzaju zatrudniają najwięcej pracowników?
- Jakie specjalizacje cieszą się największą popularnością (% zatrudnionych w stosunku do całości)?
- Kiedy (w skali roku) dokonuje się najwięcej zatrudnień pracowników? Czy zatrudnia się średnio tyle samo ludzi w skali całego roku?
- Jaki jest trend jeśli chodzi o średnie zarobki? Czy można się spodziewać ich wzrostu w przyszłości?
- Jak liczba lat doświadczenia w IT wpływa na stawkę? Czy zależne jest to od specjalizacji?
- Jakie są różnice w zarobkach między osobami które mają wykształcenie w IT a nie mają?
- Czy istnieje korelacja między wielkością zespołu i zarobkami pracownika?
- Osoby znające jakie języki obce zarabiają najwięcej? Jak znajomość angielskiego wpływa na zarobki?

2.3.2 Zakres analizy – badane aspekty

Poprzez odpowiedzenie na powyższe pytania powinniśmy określić czynniki które mają wpływ na stawkę programistów i w jakiej skali. Badane będą aspekty takie jak kraj świadczenia usług, doświadczenie w branży, wiek, płeć, wykształcenie, sektory najprężniej rozwijające się czy inne charakterystyki pracy takie jak główny język programowania czy wielkość zespołu.

2.3.3 Potencjalni użytkownicy projektowanej hurtowni danych

Analiza może wspierać procesy decyzyjne programistów dotyczące domagania się stosownej stawki, jak i w podejmowaniu decyzji dotyczących przyszłej edukacji, specjalizowania się w danej technologii. Z drugiej strony analiza może być również przydatna pracodawcą w celu lepszego określenia stawki proponowanej pracownikom.

3. Dane źródłowe

3.1. Źródła danych

Tabela 1.Zbiory danych źródłowych

Lp.	Plik, bazy danych	Typ	Liczba rek.	Rozmiar [MB]	Opis
1.	**employments**	.csv	~231 829	40	Baza zawierająca dane dotyczące pensji informatyków dla różnych krajów z całego świata w okresie od 2018 do 2021

3.2. Lokalizacja, dostępność danych źródłowych

Dane pochodzą ze strony: https://github.com/itstats/programmers_salaries na podstawie licencji GNU. Współtwórcy w README podają oryginalne źródła danych (bevopr.io, careerhigher.co, cvmira.com, eurojobs.com...).

3.3. Słownik danych – interpretacja

Tabela 2. Słownik atrybutów w pliku „**employments.csv**”

Plik: employments.csv				
Lp.	**Kolumna**	**Typ**	**Znaczenie**	**Uwagi**
1.	**age**	Numer	Wiek programisty	
2.	**date_of_ employment**	Data	Data zatrudnienia pracownika	
3.	**sex**	Tekstowy	Płeć programisty	
4.	**country**	Tekstowy	Kraj w którym pracuje programista	
5.	**experiance_ years_it**	numer	Doświadczenie programisty w IT w latach	
6.	**languages**	Tekstowy	Języki mówione znane przez pracownika	2 lub 1, drugi to zawsze angielski; wartości oddzielone przecinkami bez spacji
7.	**speciality**	Tekstowy	Specjalizacja pracownika, stanowisko, rodzaj pracy w dziedzinie IT	
8.	**core_ programming_ language**	Tekstowy	Nazwa języka programowania głównie używanego w pracy przez programistę	
9.	**academic_title**	Tekstowy	Nazwa najwyższego tytułu naukowego który programista posiada, nie koniecznie w IT	
10.	**education_ towards_it**	Tekstowy	Informacja, czy programista posiada wykształcenie w dziedzinie IT	

11.	**rate_per_hour**	Numer	Stawka pensji programisty za godzinę w euro	Stawki miesięczne przeliczane są w stawki dzienne i uzupełnione zostały te dane nawet jeśli programista otrzymywał pensję miesięczną
12.	**salary_monthly**	Numer	Stawka pensji w skali miesiąca w euro	Nie każda praca ma przypisaną tą wartość ze względu na inny rodzaj zatrudnienia
13.	**company_ country**	Tekstowy	Nazwa kraju w którym operuje firma zatrudniająca programistę	
14.	**company_type**	Tekstowy	Nazwa rodzaju firmy w której programista jest zatrudniony, kategorie dotyczą wielkości, wieku firmy, czy sposobu finansowania (np. z pieniędzy publicznych)	
15.	**work_form**	Tekstowy	Nazwa formy pracy programisty (zdalnie, nie zdalnie, mieszanie)	
16.	**team_size**	Numer	Wielkość zespołu w którym pracuje programista	
17.	**team_type**	Tekstowy	Nazwa rodzaju zespołu (lokalny, międzynarodowy)	
18.	**form_of_ employment**	Tekstowy	Nazwa formy zatrudnienia programisty (umowa o pracę, umowa zlecenie)	
19.	**full_time**	Tekstowy	Informacja, czy programista zatrudniony w pełnym wymiarze czasu	
20.	**paid_days_off**	Tekstowy	Informacja, czy w ramach pracy programista otrzymywał płatny urlop	
21.	**insurance**	Tekstowy	Informacja, czy programista był/jest ubezpieczony w ramach pracy	
22.	**training_ sessions**	Tekstowy	Informacja, czy programista brał udział w szkoleniu	

3.4. Ocena jakościowa danych

Wynik analizy jakościowej przeprowadzonej za pomocą programu Tableau oraz profilu danych SSIS został przedstawiony w tab. 3.

Tabela 3. Ocena jakościowa danych w pliku „**employments.csv”**

Lp.	Kolumna	Typ	Zakres	Ocena jakości
1.	**age**	Numer	[18-73]	Całościowo rozkład wartości dobry (max 10% dla jednej wartości)
2.	**date_of_employment**	Data	[01.01.2018 – 28.12.2021]	Data: dzień, miesiąc, rok; dobra jakość danych
3.	**sex**	Tekst	F, M	M – 85%; F – 15% W tym zawodzie występuje przewaga mężczyzn, co jest normalne w tym zawodzie; dobra jakość danych
4.	**country**	Tekst		9 krajów, między 3-21% dla jednego kraju
5.	**experiance_years_it**	numer	[0-30]	31 unikalnych wartości, max 6% dla jednej wartości – dobry rozkład
6.	**languages**	Tekstowy		Wymaga normalizacji; języki 1 lub 2, oddzielone przecinkami, drugi język to zawsze angielski; pojawiają się również rekordy: „english,english” – konieczna zamiana na pojedynczy język: „english”
7.	**speciality**	Tekstowy		14 unikalnych wartości, max 23% dla jednanej wartości (Backend); dobra jakość
8.	**core_programming_language**	Tekstowy		Other – 10%; 12 języków; między 1-22% dla jednej wartości; dobra jakość
9.	**academic_title**	Tekstowy		No degree – 48%; 5 unikalnych wartości; między 5-48% dla jednej wartości
10.	**education_towards_it**	Tekstowy	Yes, No	Yes – 25%
11.	**rate_per_hour**	Numer	[3-73.1]	Kwoty odpowiednie (3 Euro/h to ok 14 zł/h, co byłoby minimalną stawką w Polsce); odpowiednie wartości
12.	**salary_monthly**	Numer	[480-11696]	31% null – wartość wynika z faktu, że nie wszyscy programiści zatrudnieni byli ze stawką miesięczną
13.	**company_country**	Tekstowy		Kraje bardzo równomiernie rozłożone – każdy kraj stanowi ok 11% (9 krajów)
14.	**company_type**	Tekstowy		7 typów; między 5-26% dla jednej wartości

				Dobra jakość danych
15.	**work_form**	Tekstowy	hybrid, stationary, remote	Hybrid – 17%; stationary – 52%; remote – 31%; Dobra jakość danych
16.	**team_size**	Tekstowy	[3-30]	28 unikalnych wartości; między 1-10% dla jednej wartości
17.	**team_type**	Tekstowy	local, international	Local – 59%; internatnional – 41%;
18.	**form_of_employment**	Tekstowy	employee, contractor	Employee – 27%; contractor – 73%;
19.	**full_time**	Tekstowy	Yes, No	Yes – 66%, dobry rozkład
20.	**paid_days_off**	Tekstowy	Yes, No	Yes – 48%, dobry rozkład
21.	**insurance**	Tekstowy	Yes, No	Yes – 47%, dobry rozkład
22.	**training_sessions**	Tekstowy	Yes, No	Yes – 56%, dobry rozkład

4. Analityczne modele wielowymiarowe

4.1. Fakty podlegające analizie oraz ich miary

Tabela 4. Fakty oraz ich miary opracowywanych modeli analitycznych

Lp.	Fakt	Miary	Uwagi
1.	**Zatrudnienie programisty**	pensja na godzinę	-

4.2. Kontekst analizy faktów

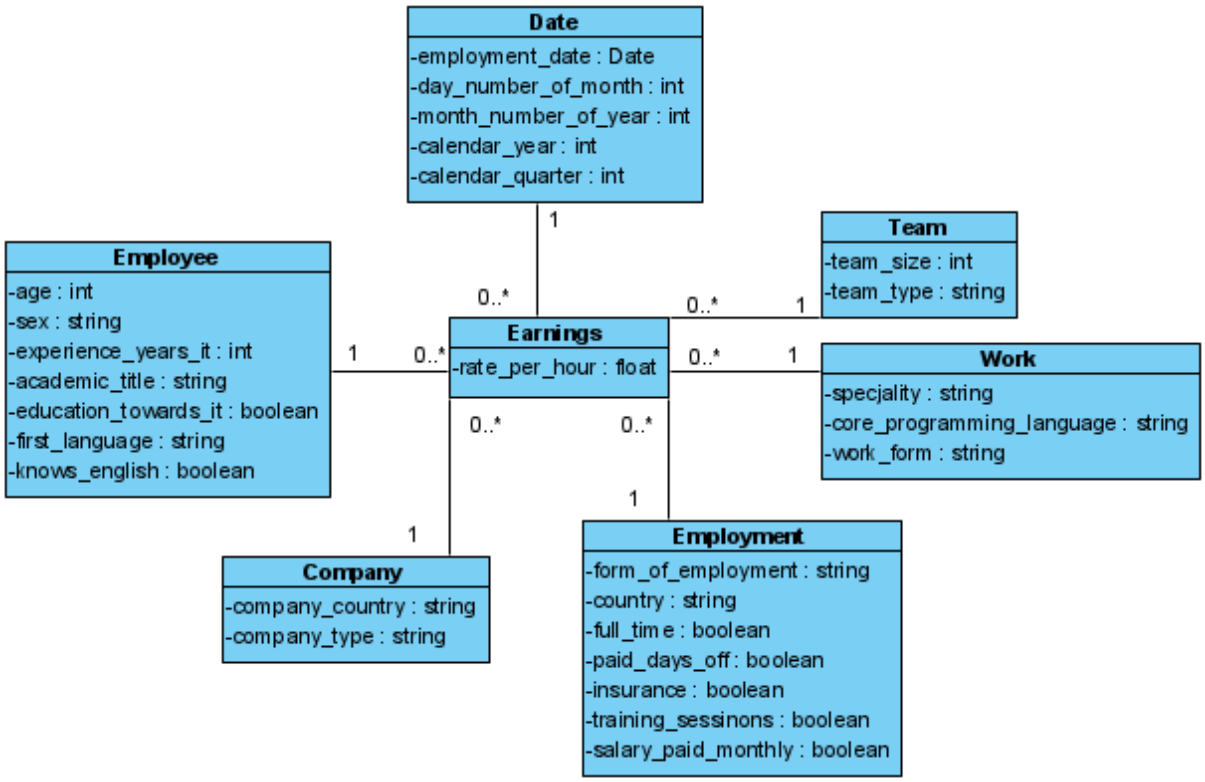
Tabela 5. Zidentyfikowane wymiary wraz z ich własnościami (charakterystykami) opracowywanych modeli analitycznych

Lp.	Kontekst analizy - wymiary	Własności
1.	**Pracownik**	Cechy programisty, jego wiek, płeć, doświadczenie, znane języki obce, wykształcenie
2.	**Praca**	Cechy charakterystyczne pracy, takie jak główny język programowania, forma pracy, specjalizacja
3.	**Zespół**	Cechy dotyczące zespołu programistycznego, jego typ, wielkość
4.	**Firma**	Cechy charakteryzujące firmę, jej rodzaj (głównie dotyczący wielkości i sposobu finansowania) jak i kraju działalności
5.	**Data zatrudniena**	Wartości dotyczące daty zatrudnienia – rok, miesiąc, kwartał, dokładna data

6.	**Zatrudnienie**	Cechy charakteryzujące zatrudnienie takie jak kraj i forma zatrudnienia, czy płatny urlop, ubezpieczenie, czy w pełnym wymiarze godzin

4.3. Modele wielowymiarowe (UML)

Proponowany model wielowymiarowy reprezentowany jest w postaci schematu gwiazdy (rysunek poniżej).



Rysunek 1. Wielowymiarowy model analityczny przedstawiony na poziomie konceptualnym

5. Projekt procesu ETL

5.1. Schemat bazy danych HD (skrypt SQL)

Baza danych została utworzona za pomocą skryptu przedstawionego w tabeli 6.

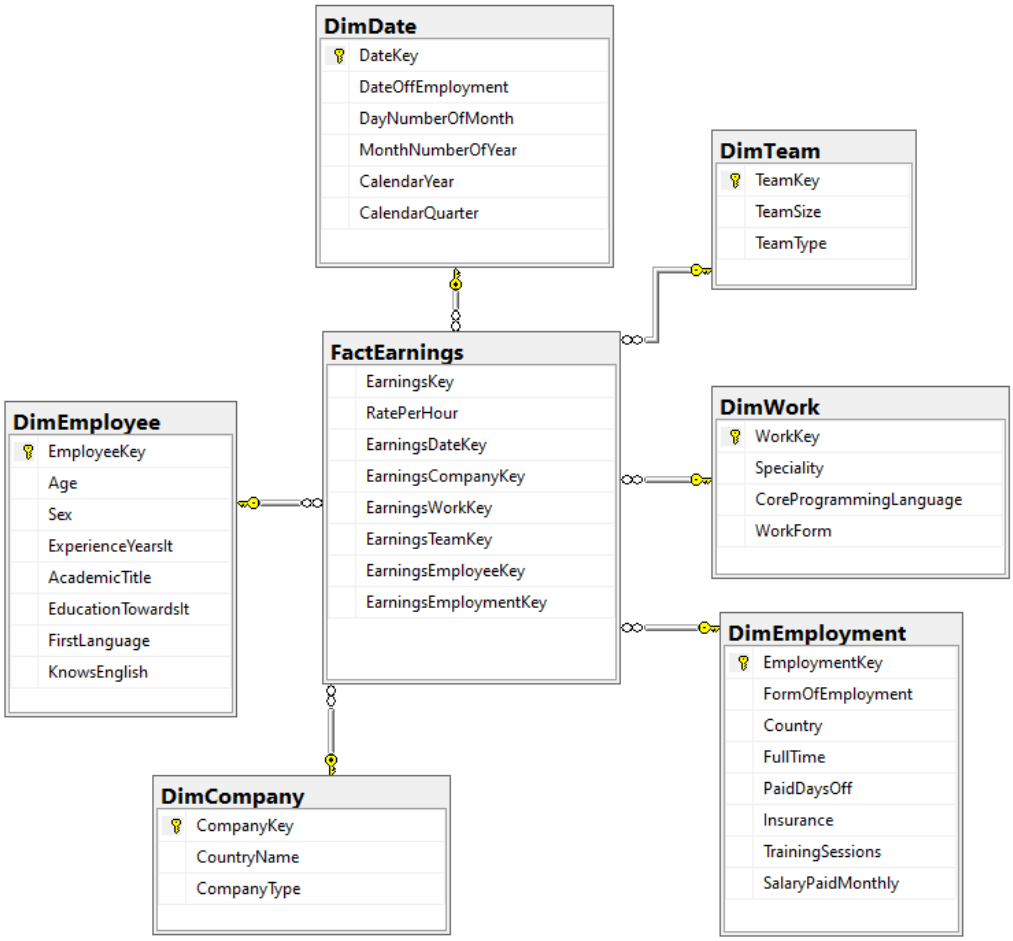
Tabela 1. Skrypt SQL tworzenia bazy hurtowni danych

```
DROP TABLE IF EXISTS FactEarnings;
DROP TABLE IF EXISTS DimCompany;
DROP TABLE IF EXISTS DimDate;
DROP TABLE IF EXISTS DimWork;
DROP TABLE IF EXISTS DimTeam;
DROP TABLE IF EXISTS DimEmployee;
DROP TABLE IF EXISTS DimEmployment;

IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='DimDate' AND XTYPE='U')
CREATE TABLE DimDate (
  DateKey INT IDENTITY(1,1) NOT NULL,
  DateOffEmployment DATE NULL,
  DayNumberOfMonth TINYINT NULL,
  MonthNumberOfYear TINYINT NULL,
  CalendarYear SMALLINT NULL,
  CalendarQuarter TINYINT NULL,

  CONSTRAINT CHK_date_not_future CHECK(DateOffEmployment IS NULL OR DateOffEmployment <=
GETUTCDATE()),
  CONSTRAINT CHK_date_not_too_far_past CHECK(YEAR(DateOffEmployment) IS NULL OR
YEAR(DateOffEmployment) > 1970),
  CONSTRAINT PK_DateKey PRIMARY KEY (DateKey),
  CONSTRAINT UNQ_DateOffEmployment UNIQUE(DateOffEmployment)
);

IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='DimCompany' AND XTYPE='U')
CREATE TABLE DimCompany (
  CompanyKey INT IDENTITY(1,1) NOT NULL,
  CountryName VARCHAR(150) NULL,
  CompanyType VARCHAR(50) NULL,

  CONSTRAINT PK_CompanyKey PRIMARY KEY (CompanyKey),
  CONSTRAINT UNQ_all_company UNIQUE(CountryName, CompanyType)
);

IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='DimWork' AND XTYPE='U')
CREATE TABLE DimWork (
  WorkKey INT IDENTITY(1,1) NOT NULL,
  Speciality VARCHAR(100) NULL,
  CoreProgrammingLanguage VARCHAR(50) NULL,
  WorkForm VARCHAR(50) NULL,

  CONSTRAINT PK_WorkKey PRIMARY KEY (WorkKey),
  CONSTRAINT UNQ_all_work UNIQUE(Speciality, CoreProgrammingLanguage, WorkForm)
);
```

```sql
IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='DimTeam' AND XTYPE='U')
CREATE TABLE DimTeam (
  TeamKey INT IDENTITY(1,1) NOT NULL,
  TeamSize SMALLINT NULL,
  TeamType VARCHAR(50) NULL,

  CONSTRAINT PK_TeamKey PRIMARY KEY (TeamKey),
  CONSTRAINT CHK_team_size_positive CHECK(TeamSize > 1),
  CONSTRAINT CHK_team_size_smaller CHECK(TeamSize < 50),
  CONSTRAINT UNQ_all_team UNIQUE(TeamSize, TeamType)
);

IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='DimEmployee' AND XTYPE='U')
CREATE TABLE DimEmployee (
  EmployeeKey INT IDENTITY(1,1) NOT NULL,
  Age SMALLINT NULL,
  Sex CHAR(1) NULL,
  ExperienceYearsIt SMALLINT NULL,
  AcademicTitle VARCHAR(50) NULL,
  EducationTowardsIt BIT NULL,
  FirstLanguage VARCHAR(50) NULL,
  KnowsEnglish BIT NULL,

  CONSTRAINT PK_EmployeeKey PRIMARY KEY (EmployeeKey),
  CONSTRAINT CHK_age_min_16 CHECK(Age >= 16),
  CONSTRAINT CHK_age_max_80 CHECK(Age <= 80),
  CONSTRAINT CHK_experience_positive CHECK(ExperienceYearsIt >= 0),
  CONSTRAINT CHK_experience_max CHECK(ExperienceYearsIt <= 60),
  CONSTRAINT UNQ_all_employee UNIQUE(Age, Sex, ExperienceYearsIt, AcademicTitle,
EducationTowardsIt, FirstLanguage, KnowsEnglish)
);

IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='DimEmployment' AND XTYPE='U')
CREATE TABLE DimEmployment (
	EmploymentKey INT IDENTITY(1,1) NOT NULL,
	FormOfEmployment VARCHAR(50) NULL,
	Country VARCHAR(50) NULL,
	FullTime BIT NULL,
	PaidDaysOff BIT NULL,
	Insurance BIT NULL,
	TrainingSessions BIT NULL,
	SalaryPaidMonthly BIT NOT NULL,

	CONSTRAINT PK_EmploymentKey PRIMARY KEY (EmploymentKey),
	CONSTRAINT UNQ_all_employment UNIQUE(FormOfEmployment, Country, FullTime,
PaidDaysOff, Insurance, TrainingSessions, SalaryPaidMonthly)
);

IF NOT EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='FactEarnings' AND XTYPE='U')
CREATE TABLE FactEarnings (
	EarningsKey INT IDENTITY(1,1) NOT NULL,
	RatePerHour MONEY NOT NULL,

	EarningsDateKey INT NOT NULL,
	EarningsCompanyKey INT NOT NULL,
	EarningsWorkKey INT NOT NULL,
	EarningsTeamKey INT NOT NULL,
	EarningsEmployeeKey INT NOT NULL,
```

```
	EarningsEmploymentKey INT NOT NULL,

	CONSTRAINT FK_DateKey FOREIGN KEY (EarningsDateKey) REFERENCES DimDate (DateKey),
	CONSTRAINT FK_CompanyKey FOREIGN KEY (EarningsCompanyKey) REFERENCES DimCompany
(CompanyKey),
	CONSTRAINT FK_WorkKey FOREIGN KEY (EarningsWorkKey) REFERENCES DimWork (WorkKey),
	CONSTRAINT FK_TeamKey FOREIGN KEY (EarningsTeamKey) REFERENCES DimTeam (TeamKey),
	CONSTRAINT FK_EmployeeKey FOREIGN KEY (EarningsEmployeeKey) REFERENCES DimEmployee
(EmployeeKey),
	CONSTRAINT FK_EmploymentKey FOREIGN KEY (EarningsEmploymentKey) REFERENCES
DimEmployment (EmploymentKey),

	CONSTRAINT CHK_rate_positive CHECK(RatePerHour > 0),
	CONSTRAINT CHK_rate_max CHECK(RatePerHour <= 100),
	CONSTRAINT UNQ_all_earnings UNIQUE(RatePerHour, EarningsDateKey,
EarningsCompanyKey, EarningsWorkKey,
	EarningsTeamKey, EarningsEmployeeKey, EarningsEmploymentKey)
);
```

Wygenerowany przez Microsoft SQL Server Management Studio schemat bazy został przedstawiony na rys. 2.

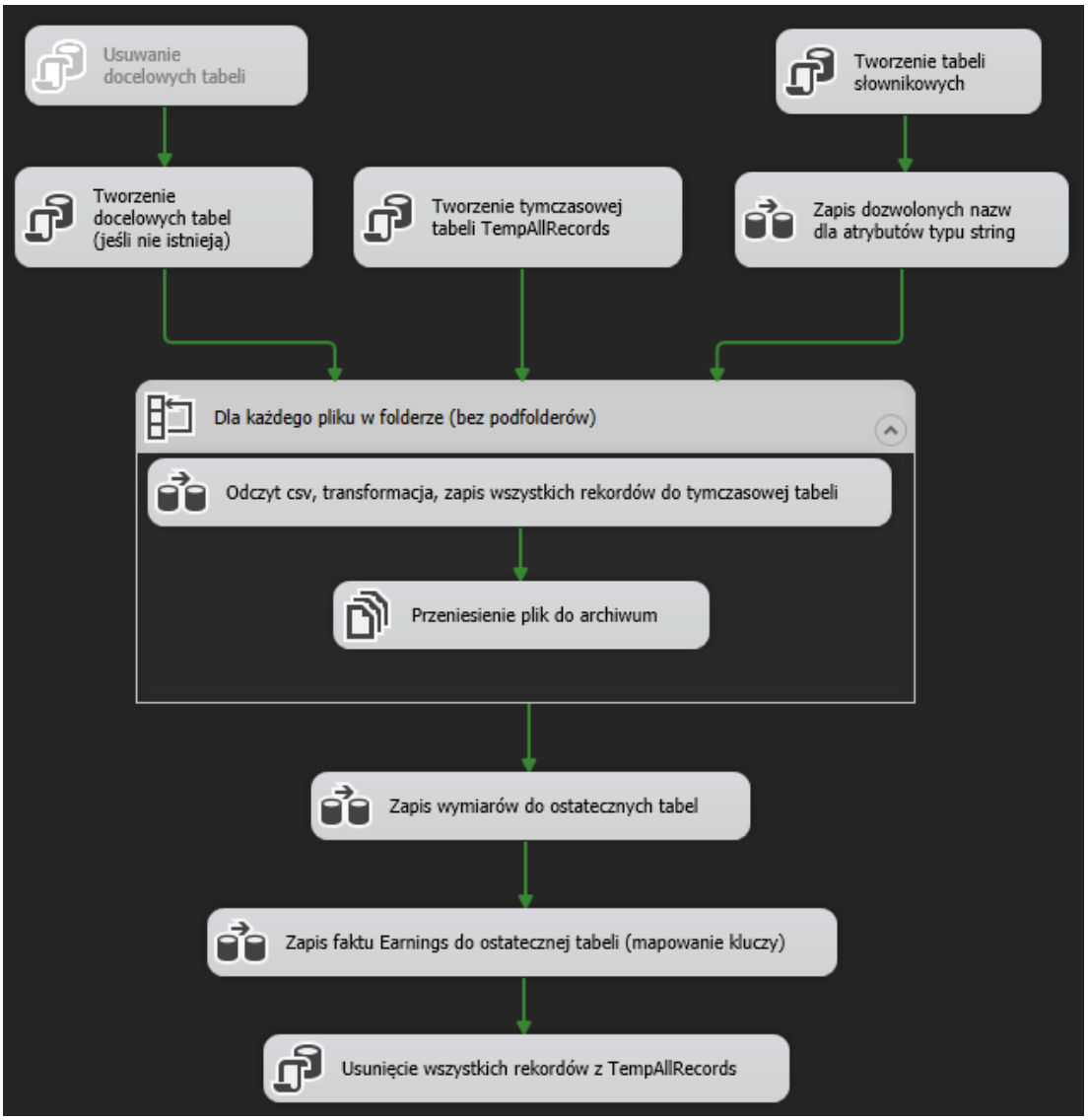


Rysunek 2. Schemat bazy danych utworzonej za pomocą skryptu (tab. 6.)

5.2. Specyfikacja procesów ETL (Control Flow + Data Flow)

Zostały zdefiniowane następujące pakiety Control Flow:

1. Execute SQL: Tworzenie docelowych tabel (jeśli nie istnieją)
2. Execute SQL: Tworzenie tymczasowej tabeli TempAllRecords
3. Execute SQL: Tworzenie tabeli słownikowych
4. Data flow: Zapis dozwolonych nazw dla atrybutów typu string
5. For each loop container: Dla każdego pliku w folderze (bez podfolderów)
    a. Data flow: Odczyt csv, transformacja, zapis wszystkich rekordów do tymczasowej tabeli
    b. File system task: Przeniesienie pliku do archiwum
6. Data flow: Zapis wymiarów do ostatecznych tabel
7. Data flow: Zapis faktu Earnings do ostatecznej tabeli (mapowanie kluczy)
8. Execute SQL: Usunięcie wszystkich rekordów z TempAllRecords



Rysunek 3. Przepływ danych (ang. Control flow)

1. Execute SQL: Tworzenie docelowych tabel (jeśli nie istnieją)

Skrypt tworzący tabele przedstawiony został w punkcie Schemat bazy danych HD

2. Execute SQL: Tworzenie tymczasowej tabeli TempAllRecords

Tabela 2. Skrypt tworzący tabele tymczasowe w hurtowni danych

```
DELETE FROM TempAllRecords;
DROP TABLE IF EXISTS TempAllRecords;
CREATE TABLE TempAllRecords (
       TempKey INT IDENTITY(1,1) NOT NULL,
    [date_of_employment] DATE,
    [age] SMALLINT,
    [sex] VARCHAR(1),
    [country] VARCHAR(50),
    [experience_years_it] SMALLINT,
    [languages] VARCHAR(50),
    [speciality] VARCHAR(50),
    [core_programming_language] VARCHAR(50),
    [academic_title] VARCHAR(50),
    [education] VARCHAR(50),
    [education_towards_it] BIT,
    [rate_per_hour] MONEY,
    [salary_monthly] VARCHAR(50),
    [company_country] VARCHAR(50),
    [company_type] VARCHAR(50),
    [work_form] VARCHAR(50),
    [team_size] SMALLINT,
    [team_type] VARCHAR(50),
    [form_of_employment] VARCHAR(50),
    [full_time] BIT,
    [paid_days_off] BIT,
    [insurance] BIT,
    [training_sessions] BIT,
    [data_source] VARCHAR(50),
    [FirstLanguage] VARCHAR(50),
    [KnowsEnglish] BIT,
    [SalaryPaidMonthly] BIT NOT NULL,
    [CalendarYear] SMALLINT,
    [MonthNumberOfYear] TINYINT,
    [DayNumberOfMonth] TINYINT,
    [CalendarQuarter] TINYINT,

       CONSTRAINT CHK_temp_team_size_positive CHECK([team_size] > 1),
       CONSTRAINT CHK_temp_team_size_smaller CHECK([team_size] < 50),
       CONSTRAINT CHK_temp_age_min_16 CHECK(Age >= 16),
       CONSTRAINT CHK_temp_age_max_80 CHECK(Age <= 80),
       CONSTRAINT CHK_temp_experience_positive CHECK([experience_years_it] >= 0),
       CONSTRAINT CHK_temp_experience_max CHECK([experience_years_it] <= 60),
       CONSTRAINT CHK_temp_rate_positive CHECK([rate_per_hour] > 0),
       CONSTRAINT CHK_temp_rate_max CHECK([rate_per_hour] <= 100),
       CONSTRAINT CHK_temp_date_not_future CHECK([date_of_employment] IS NULL OR
[date_of_employment] <= GETUTCDATE()),
       CONSTRAINT CHK_temp_not_too_far_past CHECK(YEAR([date_of_employment]) IS NULL OR
YEAR([date_of_employment]) > 1970),
```

```
	CONSTRAINT CHK_temp_date_not_too_far_past CHECK([CalendarYear] IS NULL OR
[CalendarYear] > 1970),
);
```

Do tabeli tymczasowej zostają dodane wszystkie atrybuty z oryginalnego pliku + atrybuty wyprowadzone. Dodatkowo zapewnione jest tu sprawdzanie czy atrybuty numeryczne i daty spełniają dziedziny.

3. Execute SQL: Tworzenie tabeli słownikowych

Tabela 3. Skrypt tworzenia nowych tabeli słownikowych i dodanie do ich wartości NULL

```
DROP TABLE IF EXISTS LibraryAcademicTitle;
DROP TABLE IF EXISTS LibraryCompanyType;
DROP TABLE IF EXISTS LibraryCoreProgrammingLanguage;
DROP TABLE IF EXISTS LibraryCountryName;
DROP TABLE IF EXISTS LibrarySex;
DROP TABLE IF EXISTS LibraryFirstLanguage;
DROP TABLE IF EXISTS LibraryWorkForm;
DROP TABLE IF EXISTS LibrarySpeciality;
DROP TABLE IF EXISTS LibraryTeamType;
DROP TABLE IF EXISTS LibraryFormOfEmployment;

CREATE TABLE [LibraryAcademicTitle] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryCompanyType] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryCoreProgrammingLanguage] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryCountryName] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibrarySex] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryFirstLanguage] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryWorkForm] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibrarySpeciality] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryTeamType] (
    [Name] VARCHAR(50)
)
CREATE TABLE [LibraryFormOfEmployment] (
    [Name] VARCHAR(50)
)
INSERT INTO [dbo].[LibraryAcademicTitle] VALUES (NULL);
INSERT INTO [dbo].[LibraryCompanyType] VALUES (NULL);
INSERT INTO [dbo].[LibraryCoreProgrammingLanguage] VALUES (NULL);
```

```
INSERT INTO [dbo].[LibraryCountryName] VALUES (NULL);
INSERT INTO [dbo].[LibraryFirstLanguage] VALUES (NULL);
INSERT INTO [dbo].[LibraryFormOfEmployment] VALUES (NULL);
INSERT INTO [dbo].[LibrarySex] VALUES (NULL);
INSERT INTO [dbo].[LibrarySpeciality] VALUES (NULL);
INSERT INTO [dbo].[LibraryTeamType] VALUES (NULL);
INSERT INTO [dbo].[LibraryWorkForm] VALUES (NULL);
```

Tabele słownikowe są potrzebne do sprawdzania poprawności atrybutów typu string. Dla większości ograniczone zostały one do wartości dostępnych w pliku (np. TeamType może mieć jedynie international albo local), jednak w przypadku krajów i języków dodane zostały rekordy spoza tych pojawiających się w oryginalnym pliku.

Ze względu na zezwolenie na wartości typu NULL dla wszystkich atrybutów typu string dodane zostałe one osobno do słownika (INSERTy na końcu skryput)

Ponieważ słowniki tworzone są od nowa przy każdej kompliacji, a sczytywanie jest z zewnętrznych plików tekstowych to w przypadku braku nazwy danego kraju/języka możliwe jest jej łatwe uzupełnienie.

4. Pakiet Data flow: Zapis dozwolonych nazw dla atrybutów typu string

Rysunek 4. Pakiet Data flow: Zapis dozwolonych nazw dla atrybutów typu string

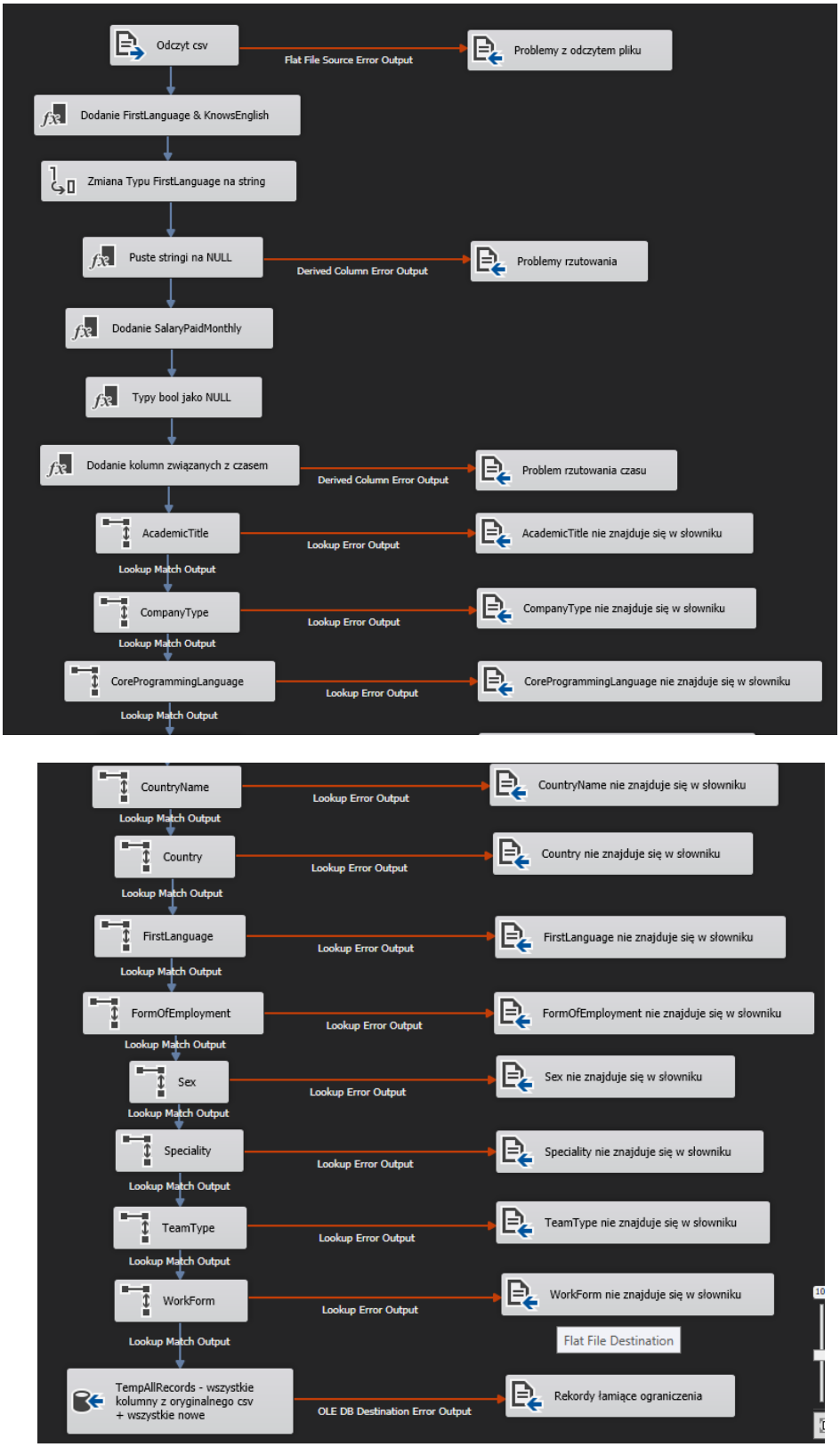
Zapis dozwolonych nazw z plików txt do baz danych w celu umożliwienia wykonania lookupa na nazwach i wykrycia błędnych rekordów.

5. For each loop container: Dla każdego pliku w folderze (bez podfolderów)

   Pliki typu .csv o dowolnej nazwie trzeba dodać do folderu w celu ich zaczytania.

   a. Pakiet Data flow: Odczyt csv, transformacja, zapis wszystkich rekordów do tymczasowej tabeli

Wykonanie wyprowadzeń atrybutów (FirstLanguage, KnowsEnglish), sprawdzenie czy atrybuty typu string są poprawne (lookup z wcześniej wczytanymi słownikami) oraz zapisanie rekordów do tabeli tymczasowej. Ze względu na ograniczenia nałożone na tabelę TempAllRecords do tego miejsca mamy zapewnioną poprawność wszystkich rekordów (nie ma zapewnionego braku powtarzalności, jedynie poprawność).

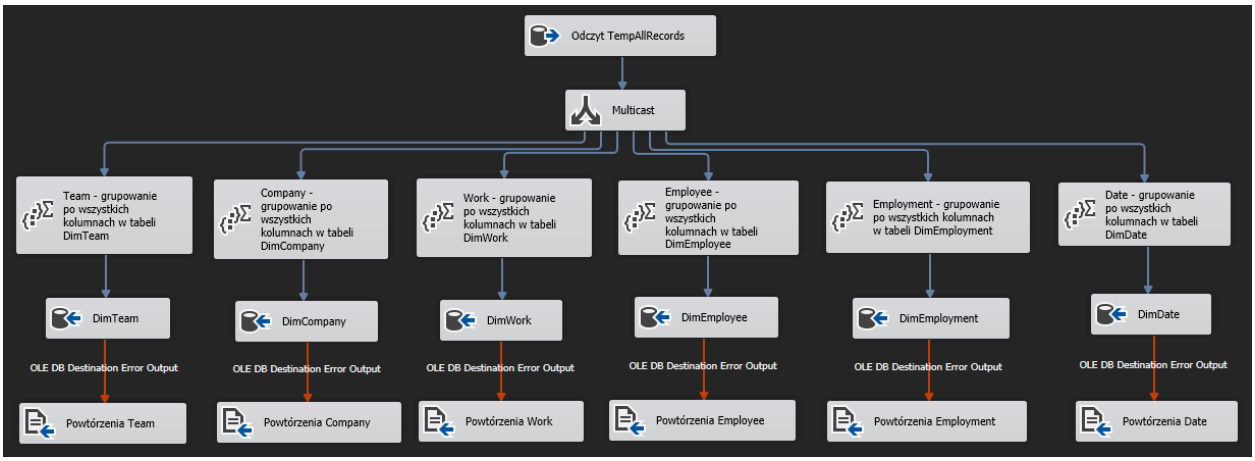


Rysunek 5. Pakiet data flow: Odczyt csv, transformacja, zapis wszystkich rekordów do tymczasowej tabeli

b. File system task: Przeniesienie pliku do archiwum

Przeniesienie pliku wczytanego do podfolderu „archive” znajdującego się w folderu źródłowym.

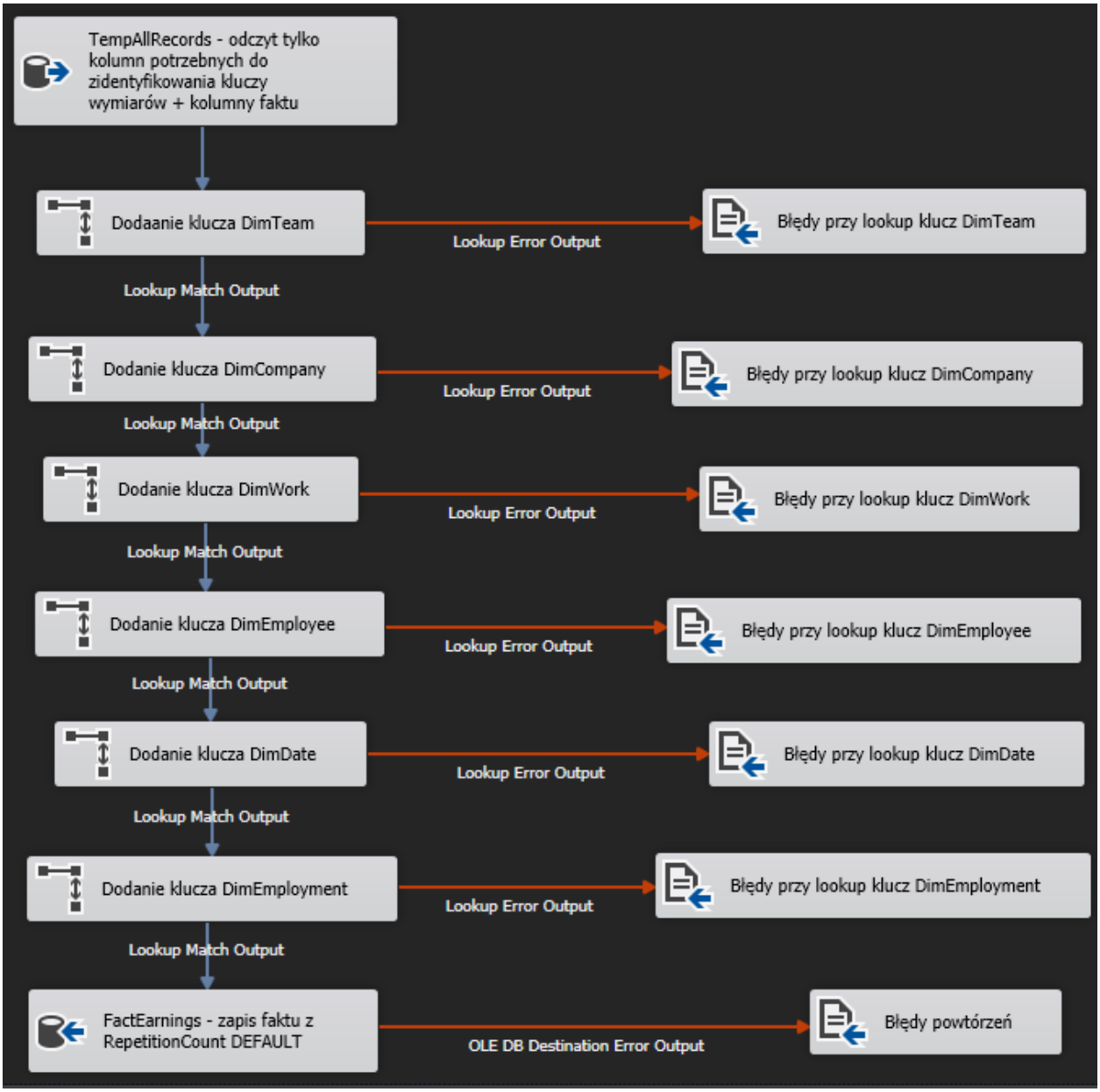
6. Pakiet data flow: Zapis wymiarów do ostatecznych tabel



Rysunek 6. Pakiet data flow: Zapis wymiarów do ostatecznych tabel

W celu zapewnienia braku powtarzalności wykonywany zostaje na rekordach najpierw AGREGATE (GROUP BY dla każdego atrybutu), jako że jednak dany wymiar może znajdować się już w bazie przy tworzeniu tabel dodane zostały ograniczenia typu UNIQUE (np. `UNQ_all_employment` w tabeli DimEmployment) które zapewniają brak powtarzalności wymiarów.

7. Pakiet data flow: Zapis faktu Earnings do ostatecznej tabeli (mapowanie kluczy)



Rysunek 7. Zapis faktu Earnings do ostatecznej tabeli (mapowanie kluczy)

Dla każdego zestawu atrybutów dotyczącego danego wymiaru zostaje wykonany lookup w celu uzyskania klucza obcego dla tego wymiaru, następnie dodawane są wszystkie rekordy do tabeli faktu wraz z uzyskanymi kluczami.

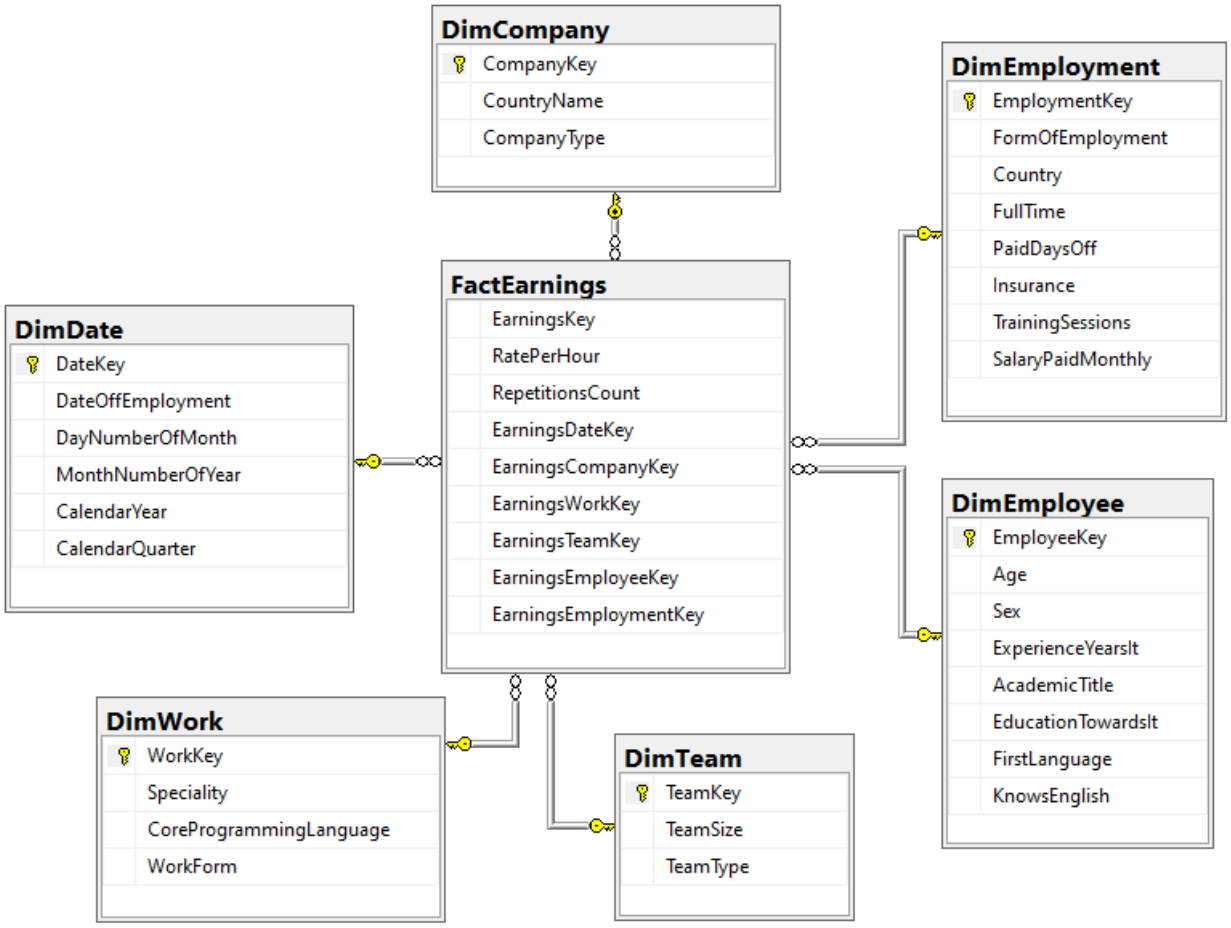
Na tabelę faktu nałożone jest ograniczenie UNIQUE `UNQ_all_earnings` uniemożliwiające dodanie dwóch identycznych rekordów, w tym miejscu więc uzyskujemy listę z ewentualnymi powtórzeniami – jeśli spróbujemy załadować drugi raz ten sam plik zostanie to również wykryte w ramach tej listy.

8. Execute SQL: Usunięcie wszystkich rekordów z TempAllRecords

Ponieważ usunięcie tabel tymczasowych powoduje problemy z ponownym uruchomieniem Integration Services Project a jednocześnie rekordy tymczasowe zajmują niepotrzebnie miejsce, to zdecydowałam się na pozostawienie pustej tabeli tymczasowej.

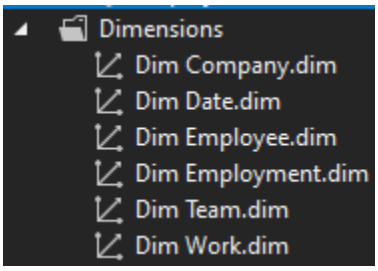
6. Implementacja modeli wielowymiarowych

6.1. Widok danych



Rysunek 7. Diagram przedstawiający ostateczne wymiary i fakt po dodaniu rekordów.

6.2. Wymiary



Rysunek 8. Diagram przedstawiający ostateczne wymiary i fakt po dodaniu rekordów.

Dodane Named Calculation dla wymiarów:

1. Dim Employment

a) AgeGroup – podział co 10 lat do 70+

Tabela 4. Podział wieku pracowników na grupy

```
CASE
	WHEN Age IS NULL THEN 'Unknown'
	WHEN Age <= 20 THEN '10s'
	WHEN Age <= 30 THEN '20s'
	WHEN Age <= 40 THEN '30s'
	WHEN Age <= 50 THEN '40s'
	WHEN Age <= 60 THEN '50s'
	WHEN Age <= 70 THEN '60s'
	ELSE '70+'
END
```

b) ExperienceYearsItGroup – podział co 5 lat do 30+

Tabela 5. Podział liczby lat doświadczenia pracownika na przedziały

```
CASE
	WHEN ExperienceYearsIt IS NULL THEN 'Unknown'
	WHEN ExperienceYearsIt < 5 THEN 'Less than 5'
	WHEN ExperienceYearsIt < 10 THEN 'Less than 10'
	WHEN ExperienceYearsIt < 15 THEN 'Less than 15'
	WHEN ExperienceYearsIt < 20 THEN 'Less than 20'
	WHEN ExperienceYearsIt < 25 THEN 'Less than 25'
	WHEN ExperienceYearsIt < 30 THEN 'Less than 30'
	ELSE '30+'
END
```

2. Dim Team

a) TeamSizeGroup – przedziały mniej schematyczne niż powyżej

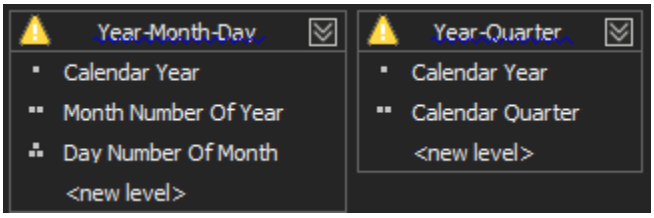
Tabela 6. Podział wielkości grup na przedziały

```
CASE
	WHEN TeamSize IS NULL THEN 'Unknown'
	WHEN TeamSize < 7 THEN 'Small'
	WHEN TeamSize < 15 THEN 'Middle size'
```

```
        WHEN TeamSize < 25 THEN 'Big'
        ELSE 'Huge'
END
```

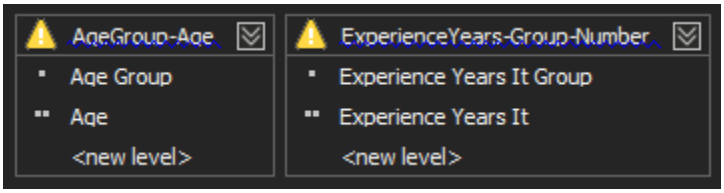
Dodane hierarchie dla wymiarów:

1. Dim Date



Rysunek 8. Hierarchie dla wymiaru Dim Date

2. Dim Employee

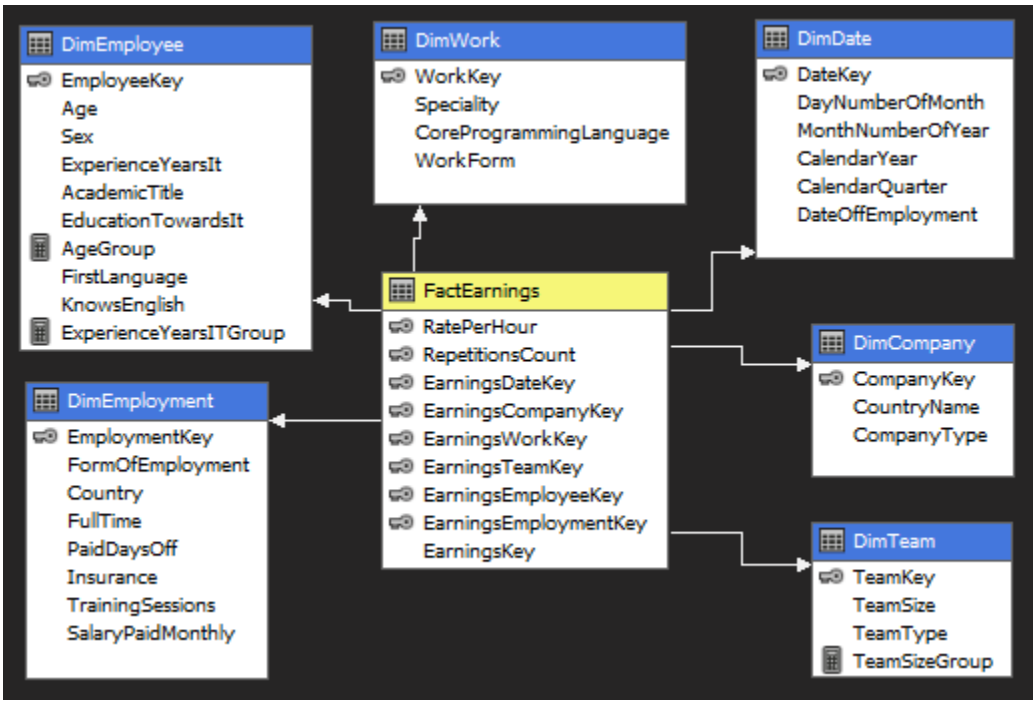


Rysunek 9. Hierarchie dla wymiaru Dim Employee

3. Dim Team

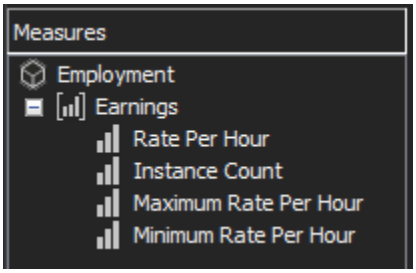
Rysunek 10. Hierarchia dla wymiaru Dim Team

6.3. Modele wielowymiarowe – Kostki



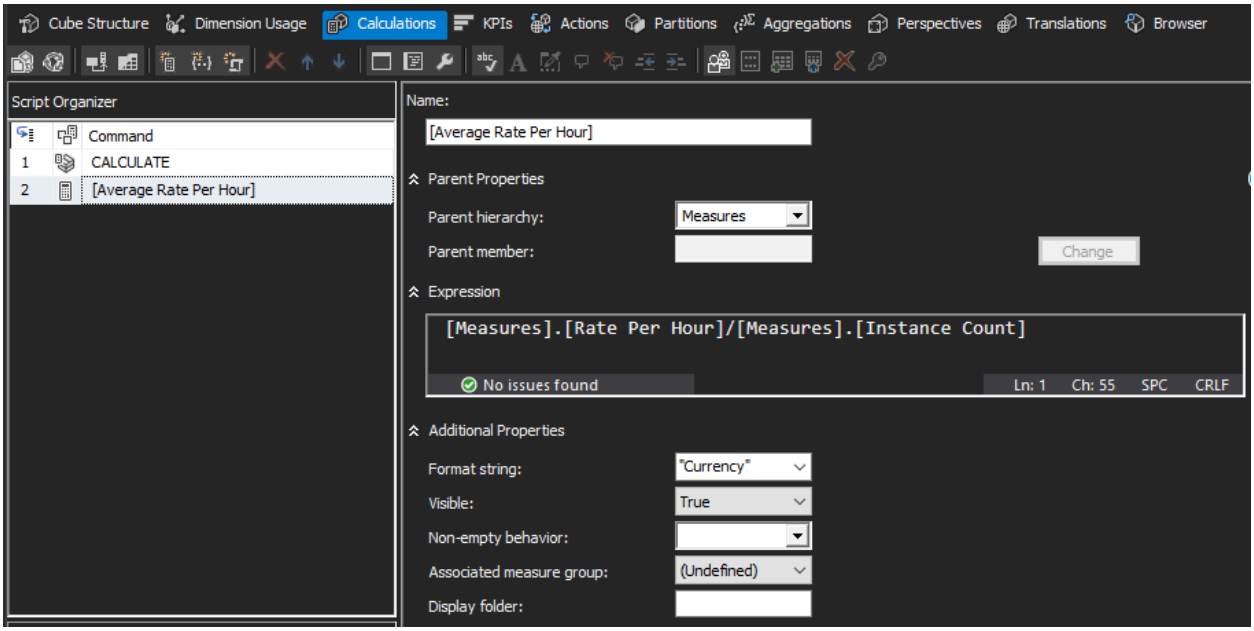
Rysunek 11. Ostateczna cube structure dla projektu wielowymiarowego

Miary dla faktu:



Rysunek 12. Miary dla faktu Fact Earnings

Dodatkowo zdefiniowana została miara wyliczana (Calculations) zwracająca średnią stawkę za godzinę dla danej grupy [Average Rate Per Hour].

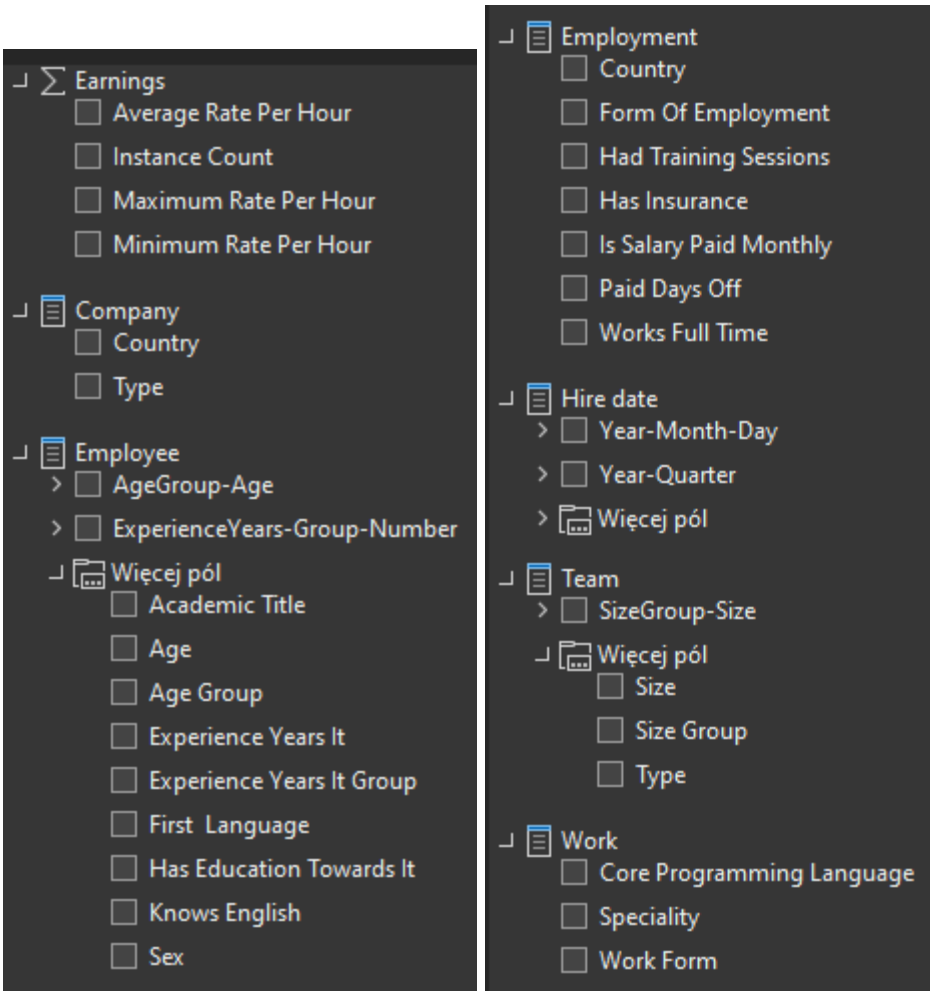


Rysunek 13. Miara [Average Rate Per Hour] wyliczana na podstawie sumy stawek za godzinę i liczby rekordów.

Dodatkowo dla ułatwienia pracy dla wszystkich kluczy wymiarów oraz dla miary [Rate Per Hour] ustawiona została własność Visible na False

Rysunek 14. Ustawienie widoczności w zakładce Properties > Advanced.

Dzięki temu ustawieniu dane atrybuty nie są widoczne w pliku .xlsx

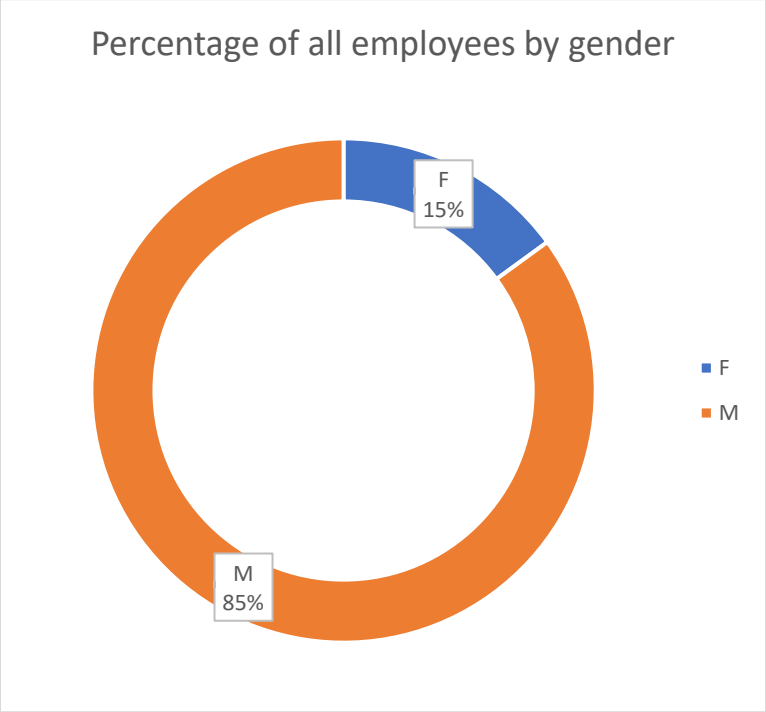


Rysunek 15 i 16. Widoczne w pliku .xlsx atrybuty, hierarchie i miary.

7. Analiza danych

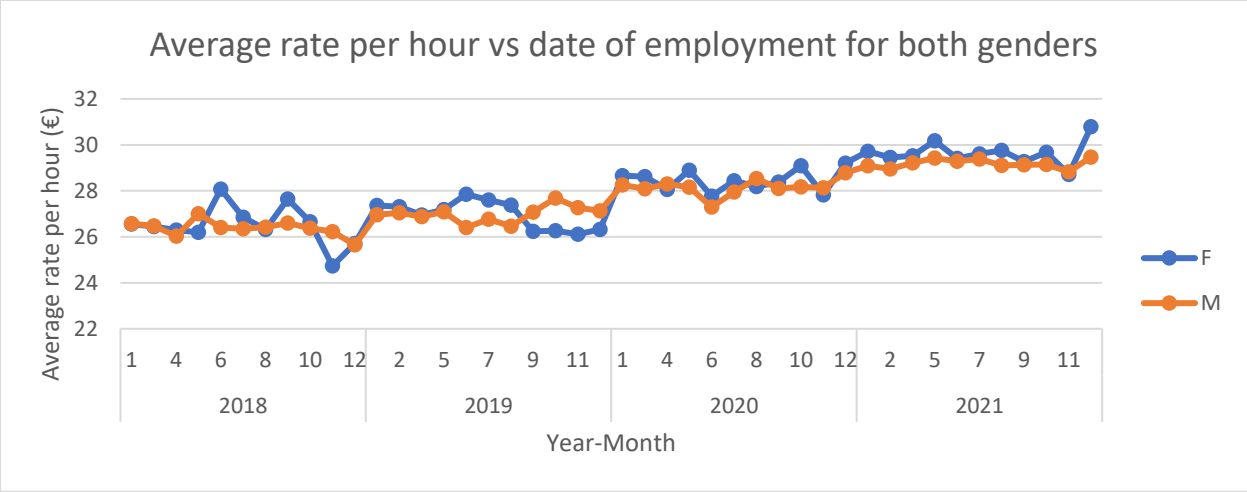
Procesy analityczne wykonane zostały w ramach dostępnego w Visual Studio pliku .xlsx z tabelami przestawnymi.

7.1. Realizacja procesów analitycznych



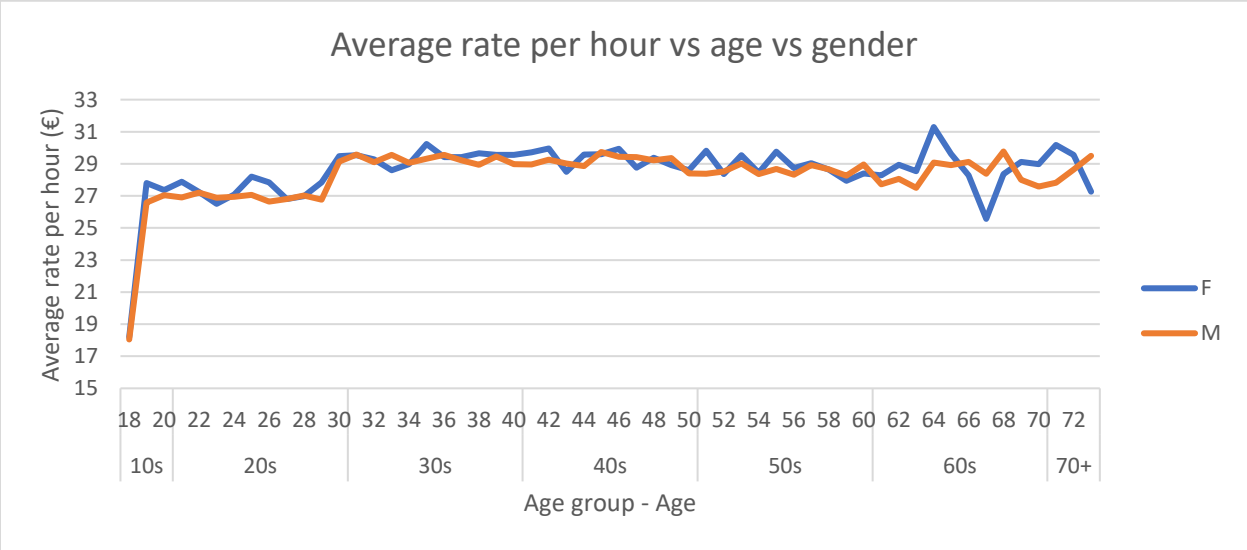
Wykres 1. Procent programistów płci męskiej w porównaniu do kobiet

Jak widać większość pracowników zatrudnionych w IT stanowią mężczyźni (85%). Ta dana sama w sobie nie jest w żaden sposób przełomowa, trzeba jednak mieć ją na uwadze przy dalszej analizie uwzględniającej różnice między płciami.



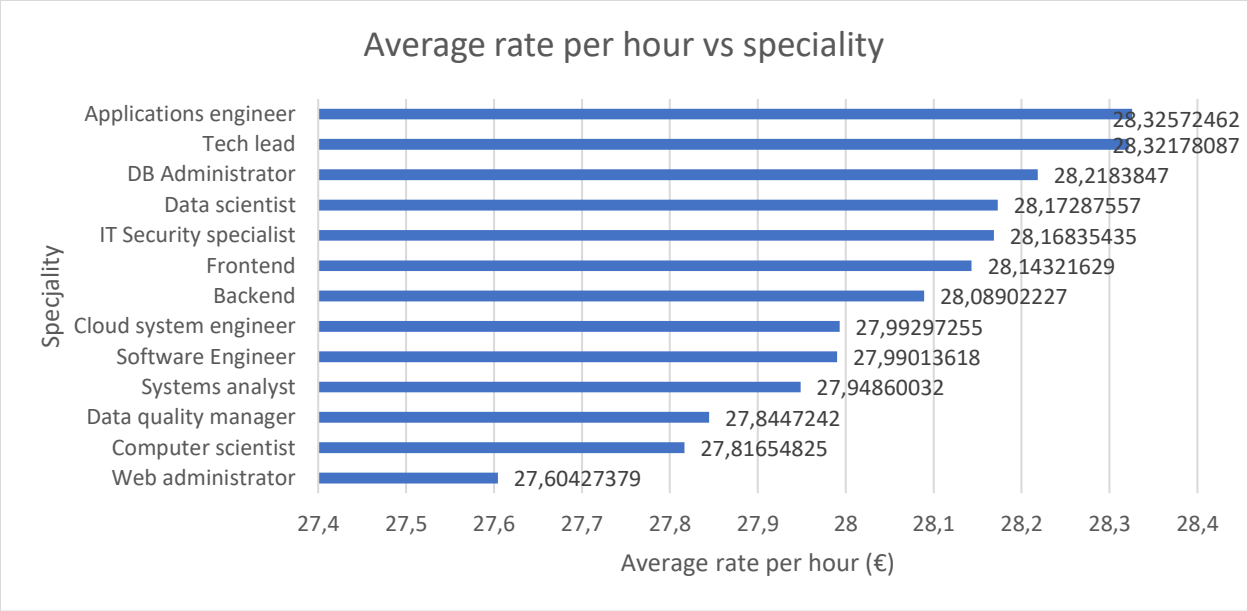
Wykres 2. Średnia stawka za godzinę w euro kobiet i mężczyzn na przestrzeni lat

Jak widać na wykresie zarobki kobiet i mężczyzn nie różnią się znacznie od siebie (max różnica 2 euro), trend dla obu jest wzrostowy.



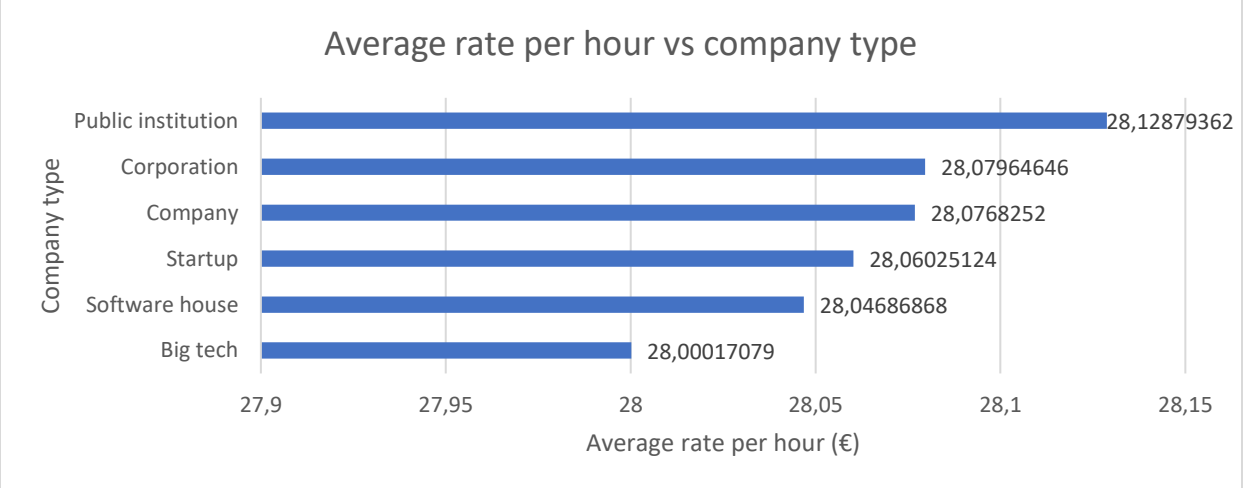
Wykres 3. Średnia stawka za godzinę w euro w zależności od wieku dla kobiet i mężczyzn

Na tym wykresie również można zaobserwować, że dla wszystkich przedziałów wiekowych z pominięciem 60-latków stawki dla obu płci są podobne.



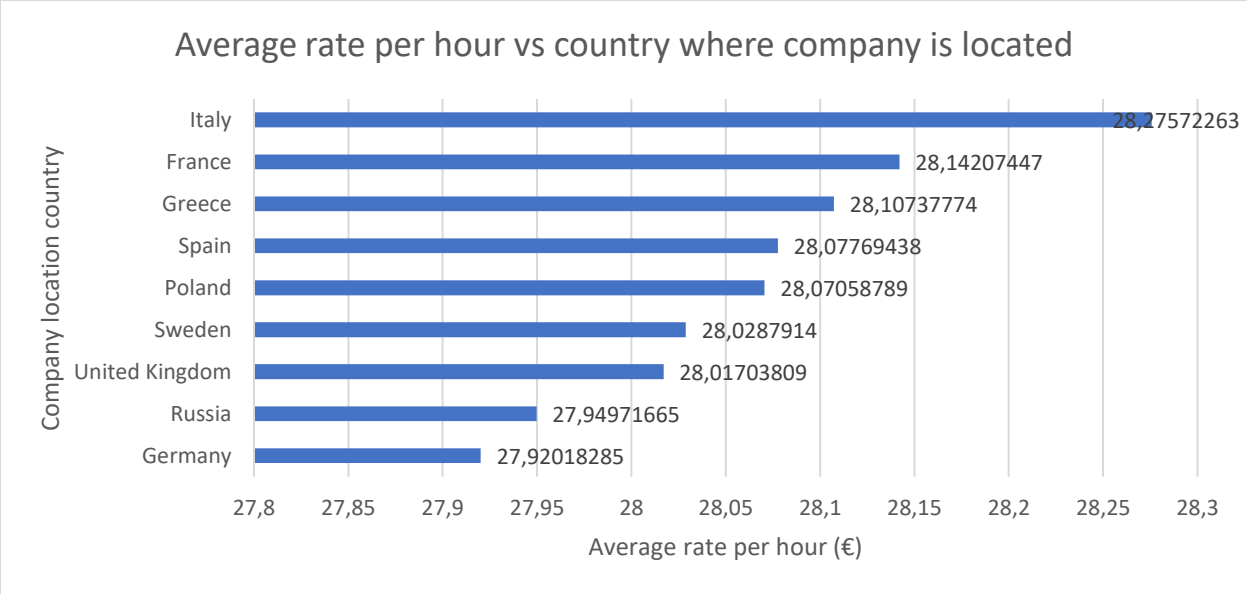
Wykres 4. Średnia stawka za godzinę w euro w zależności od specjalizacji

Różnica w średnich stawkach między poszczególnymi specjalizacjami nie jest znaczna (max ok 0,7 euro). Ciężko jest zaobserwować konkretną zależność między poszczególnymi specjalizacjami która mogłaby wyjaśnić różnice w zarobkach.



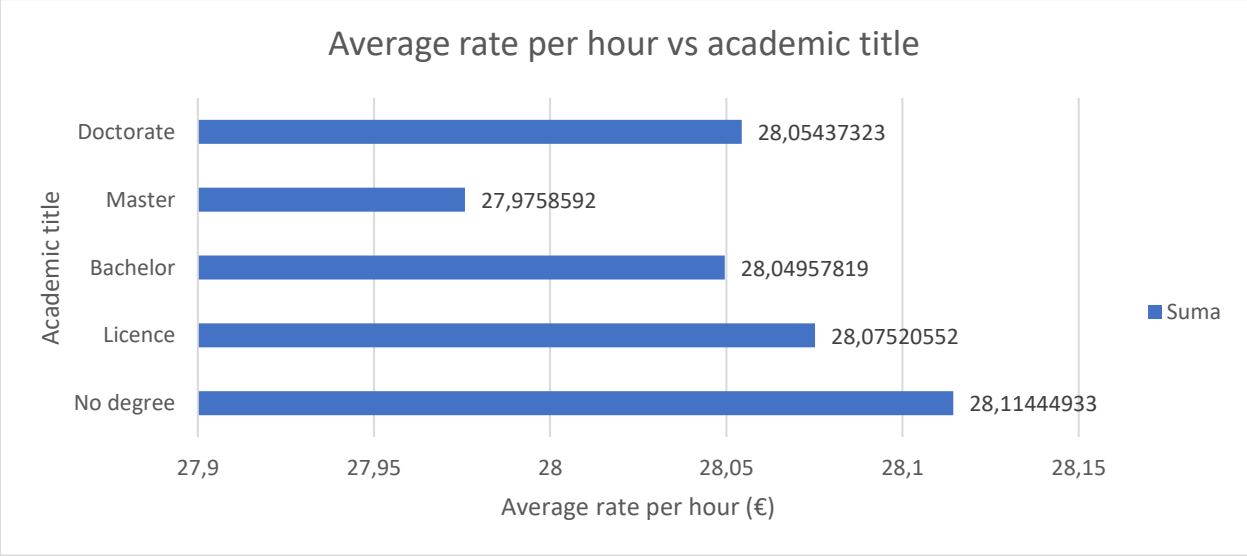
Wykres 5. Średnia stawka za godzinę w euro w zależności od typu instytucji

Niespodziewaną rzeczą w tym przypadku jest fakt, że najlepiej płacącą instytucją (średnio) nie są korporacje lecz publiczne instytucje, a najmniej Big Tech (korporacje typu Google, Facebook, Amazon) które powszechnie mają renomę w świadomości publicznej jako najlepiej płacące instytucje.



Wykres 6. Średnia stawka za godzinę w euro w zależności od kraju w którym zlokalizowana jest firma.

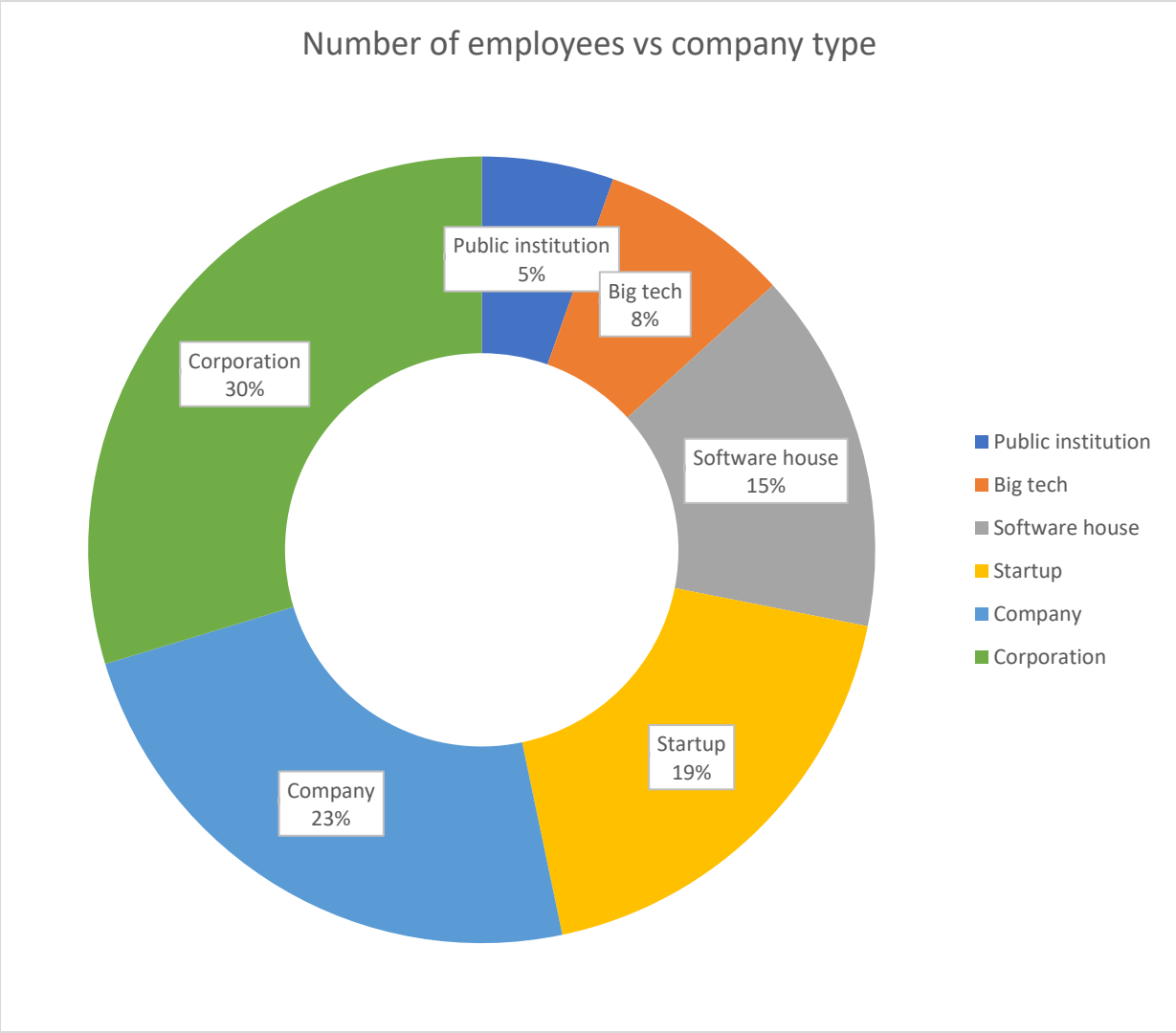
Co ciekawe liderami w przypadku stawek nie są kraje anglojęzyczne czy Niemcy, ale Włochy i Francja. Najprawdopodobniej wynika to z problemów językowych – włoski i francuski nie są tak popularnymi językami jak angielski czy niemiecki (w kraje niemieckojęzyczne wliczają się Niemcy, Austria, Szwajcaria, Luxemburg) i firmy te muszą z tego względu proponować bardziej konkurencyjne stawki.



Wykres 7. Średnia stawka za godzinę w euro w zależności od wykształcenia

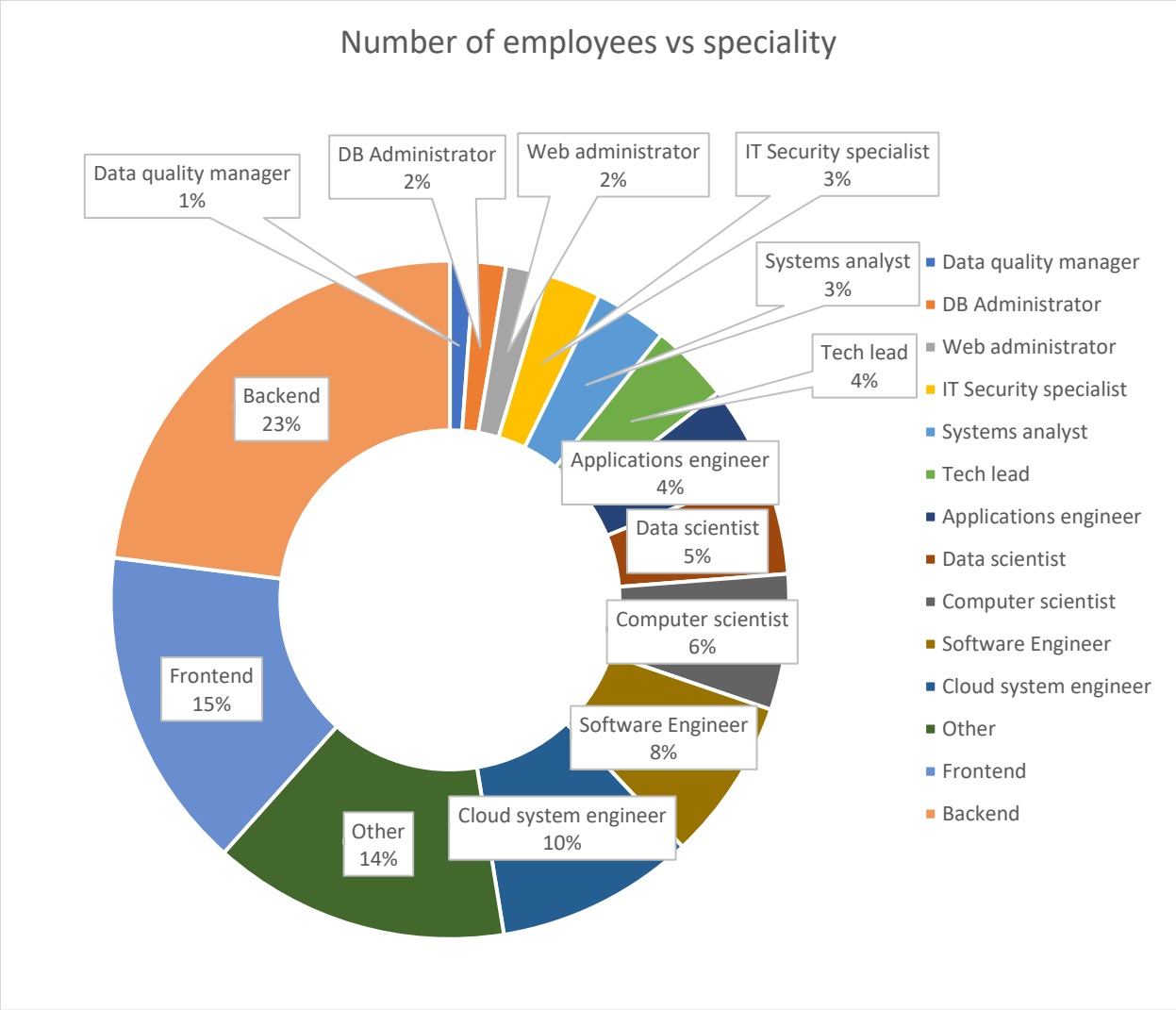
Średnio najwięcej zarabiają osoby bez wykształcenia – można to wyjaśnić faktem, że informatyk to zawód bardzo praktyczny, wymagający technicznych umiejętności których można się nauczyć w pracy.

Osoby bez wykształcenia przeważają więc doświadczeniem praktycznym które w tym przypadku przekłada się na wyższe zarobki. Trzeba tu również zauważyć, że ogólny trend jest spadkowy im wyższe wykształcenie, z pominięciem doktoratów, którzy to zarabiają więcej niż magistrzy.



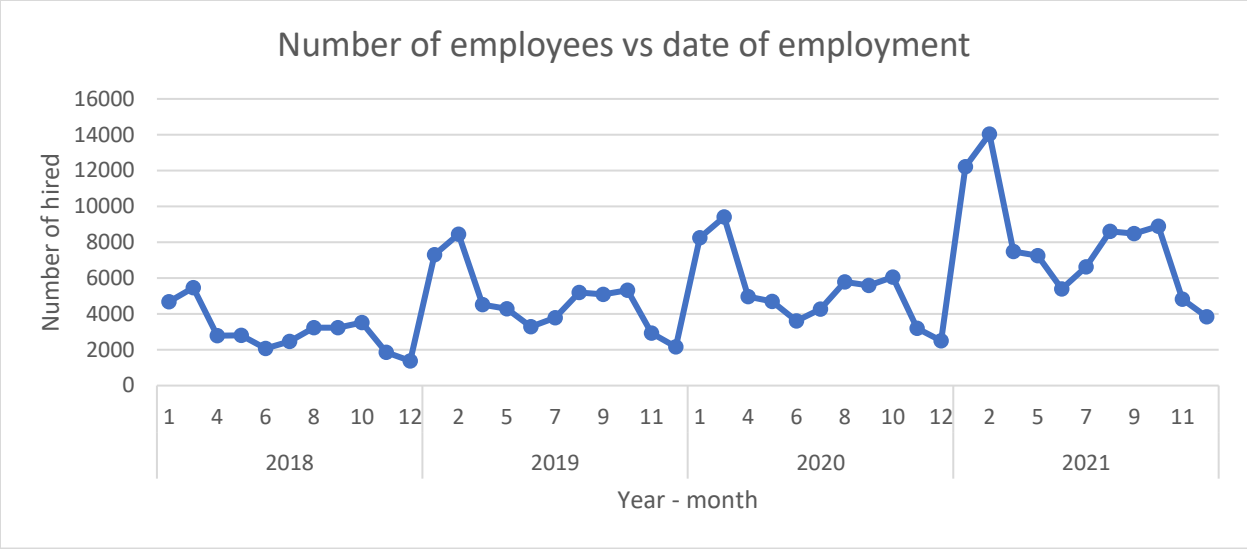
Wykres 8. Procent informatyków pracujących w firmach poszczególnego typu

W kontekście również wykresu 5 – publiczne instytucje co prawda płacą najwięcej, lecz również stanowią najmniejszy procent wszystkich instytucji zatrudniających pracowników, największy procent stanowią korporacje i firmy.



Wykres 9. Procent pracowników zatrudnionych na poszczególnych specjalizacjach.

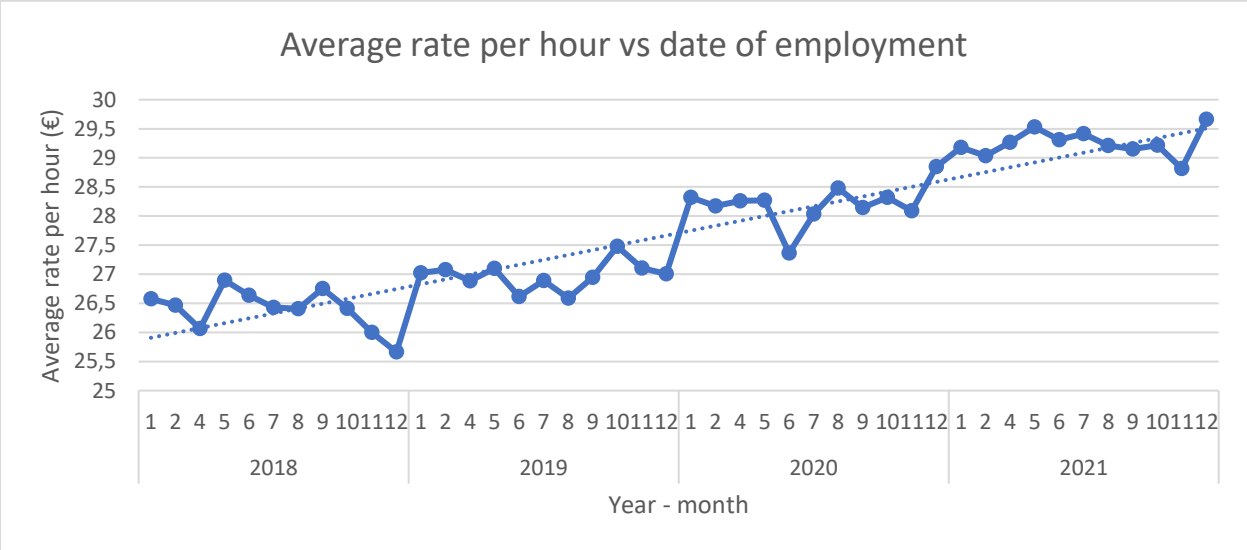
Jak widać najwięcej zatrudnionych informatyków jest w Backendzie i Frontendzie, natomiast bardziej specyficzne specjalizacje stanowią znacznie mniejszy procent całości. Co istotne w tym przypadku średnie stawki dla frontedu i backentu są po środku skali, natomiast dla pozostałych można trafić zarówno na dobrze płatne (aplications engeneer) jak i źle płatne (web administrator)



Wykres 10. Liczba zatrudnionych pracowników w czasie do 2018-2021.

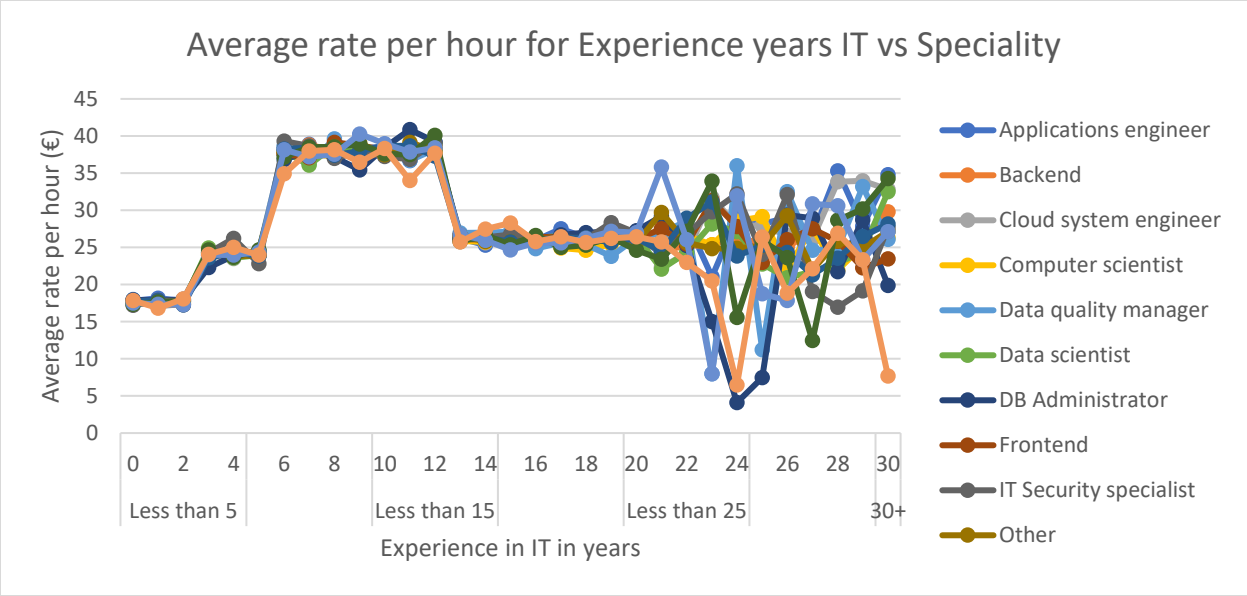
W przypadku tego wykresu dość ładnie prezentuje się trend – najwięcej zatrudnieni jest na początku roku i tuż przed jego końcem, widać też, że z każdym rokiem liczba zatrudnionych rośnie.

Gwałtowny skok w zadudnieniach na początku roku może wynikać z formy podpisywanych kontraktów – do końca roku. W związku z tym nowe umowy podpisuje się na początku roku.



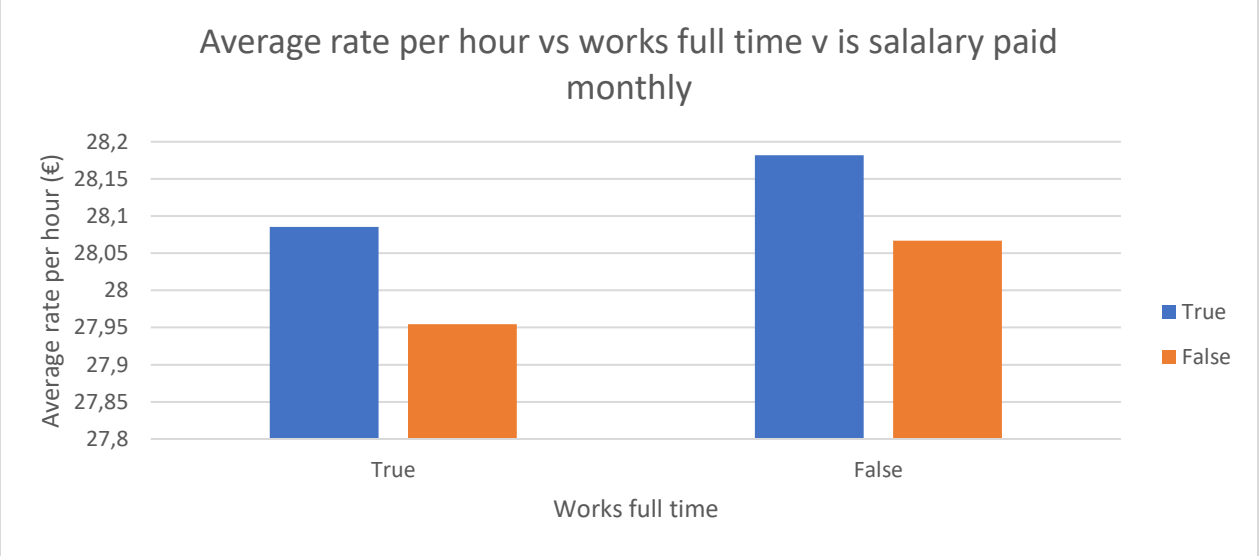
Wykres 11. Średnia stawka za godzinę w euro w zależności od czasu

Jak widać na przestrzeni lat średnie zarobki informatyków rosną w stosunkowo regularny sposób (linia trendu).



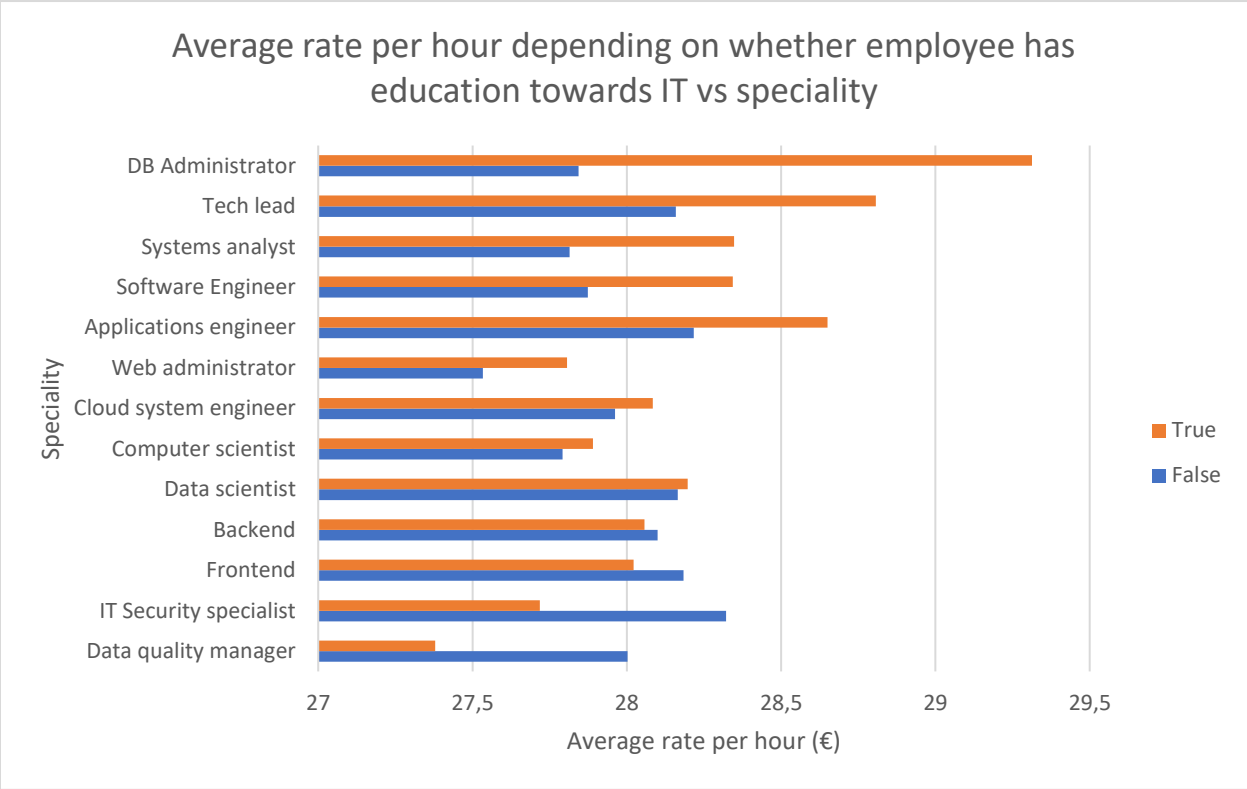
Wykres 12. Średnie stawka za godzinę w euro w zależności od lat doświadczenia dla różnych specjalizacji

Na tym wykresie prezentuje się ciekawy ogólny trend – dla wszystkich specjalizacji dla doświadczenia poniżej 20 lat średnia stawka za godzinę jest podobna dla wszystkich specjalizacji, jednak dla osób z doświadczeniem powyżej 20 lat nie ma widocznego trendu. Wynika to najprawdopodobniej z faktu, że wiele technologii się starzeje i po paru latach funkcjonowania może być albo wciąż w użytku albo nieużywana – stąd niepewność w kwestii zarobków programistów pracujących w danej technologii.



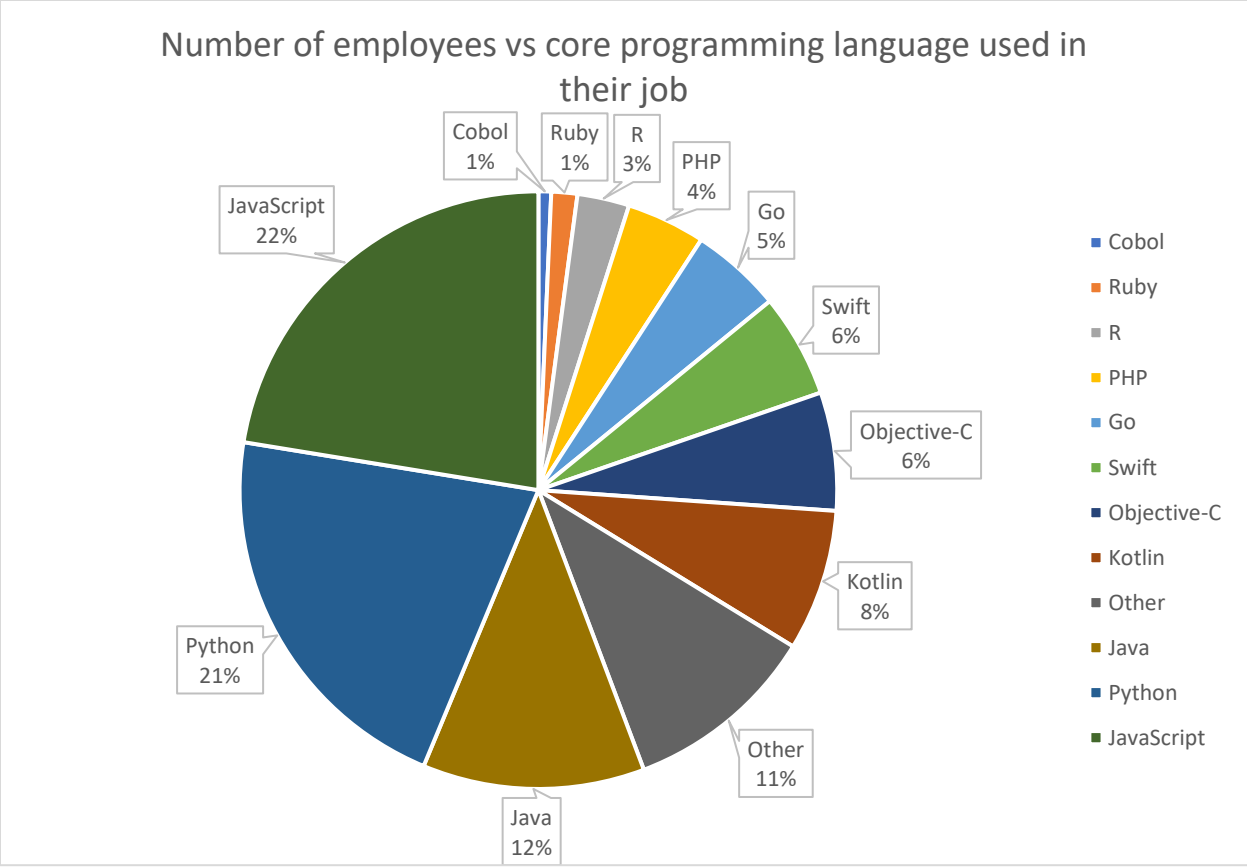
Wykres 13. Średnie stawka za godzinę w euro w zależności od faktu czy pensja wypłacana miesięcznie i czy pracownik pracuje w pełnym wymiarze godzin

Jak widać z wykresu praca w pełnym wymiarze godzin jest zwykle mniej opłacalna, jednak jeśli chodzi o sposób wypłacania stawki (miesięcznie/dziennie) to osoby ze stawką miesięczną w obu przypadkach zarabiają średnio więcej.



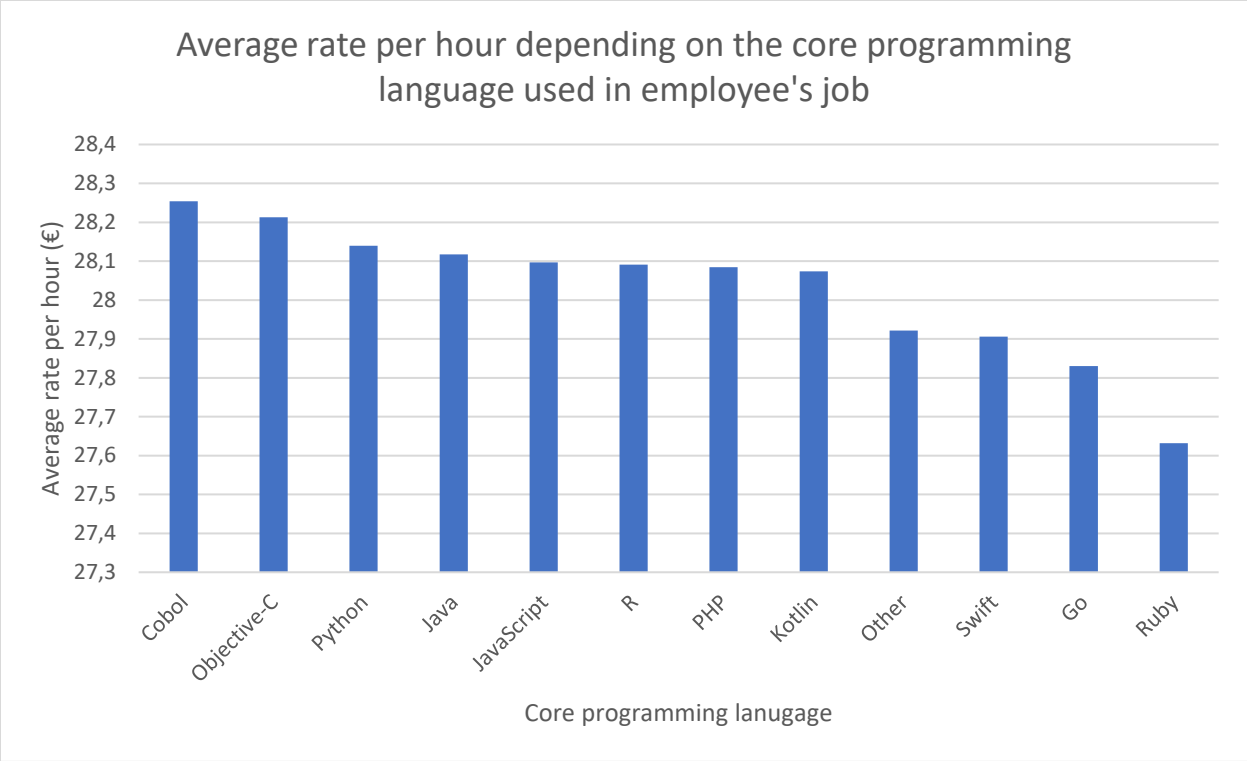
Wykres 14. Średnie stawka za godzinę w euro w zależności od faktu czy pracownik posiada wykształcenie w kierunku IT w zależności od specjalizacji

Nie we wszystkich specjalizacjach osoby z wykształceniem informatycznym zarabiają więcej, dwa najbardziej rzucające się w oczy pod tym względem to Data quality manager i IT Security specialist gdzie osoby bez wykształcenia informatycznego zarabiają więcej. Różnice tu są jednak rzędu jedynie 1.1 euro maxymalnie.



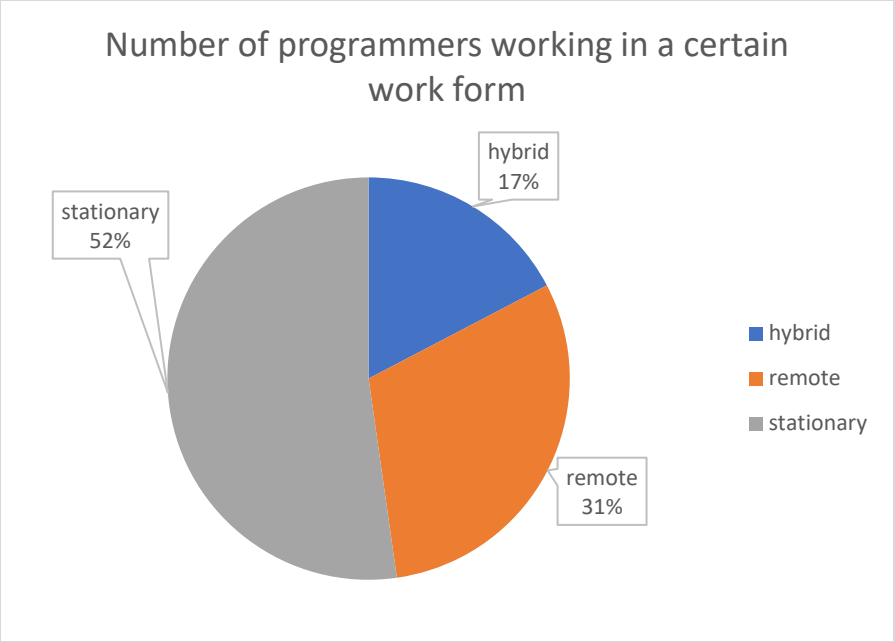
Wykres 15. Liczba osób zatrudnionych względem języka programowania głównie używanego w ich pracy.

Jak widać najwięszą popularnością cieszy się JavaScript, następnie Python i Java. Ta kolejność jest o tyle zaskakująca, że Python jest stosunkowo nowym językiem programowania i mimo tego jest bardzo dominujący na rynku. Wiele języków ma udział procentowy poniżej 10%.



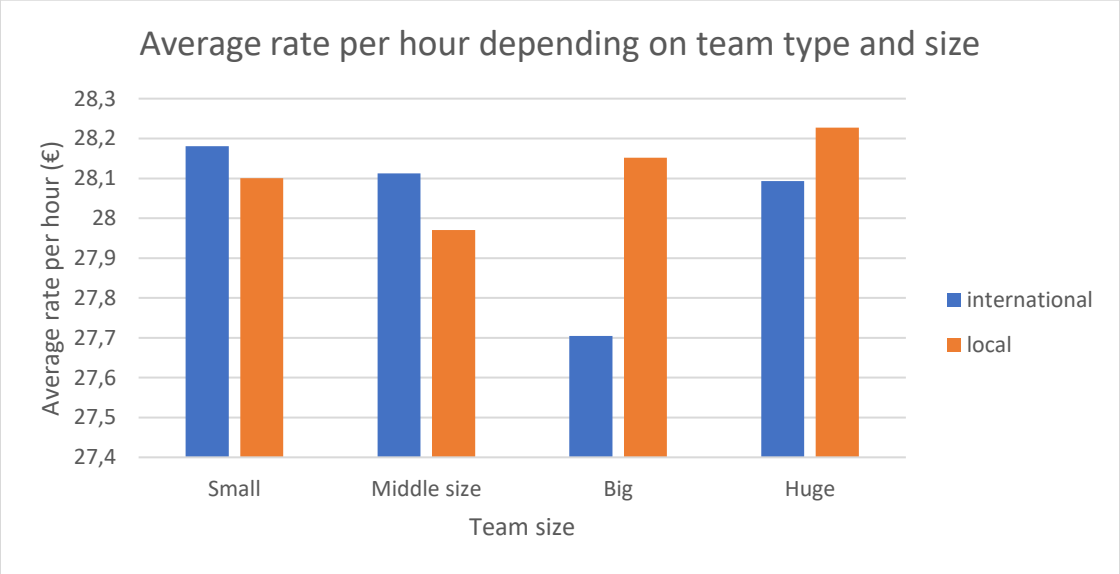
Wykres 16. Średnia stawka za godzinę w euro względem głównego języka programowania używanego w pracy programisty.

W kontekście wyników z wykresu 15 możemy zauważyć, że język z najmniejszym udziałem na rynku jest jednocześnie najbardziej opłacalnym, kolejny język (Objective-C) ma udział zaledwie 6% wszystkich zatrudnieni. Ważne jest jednak zaznaczenie, że różnice w średnich są mimo wszystko stosunkowo niewielkie (najwięcej 0.6 euro za godzinę).



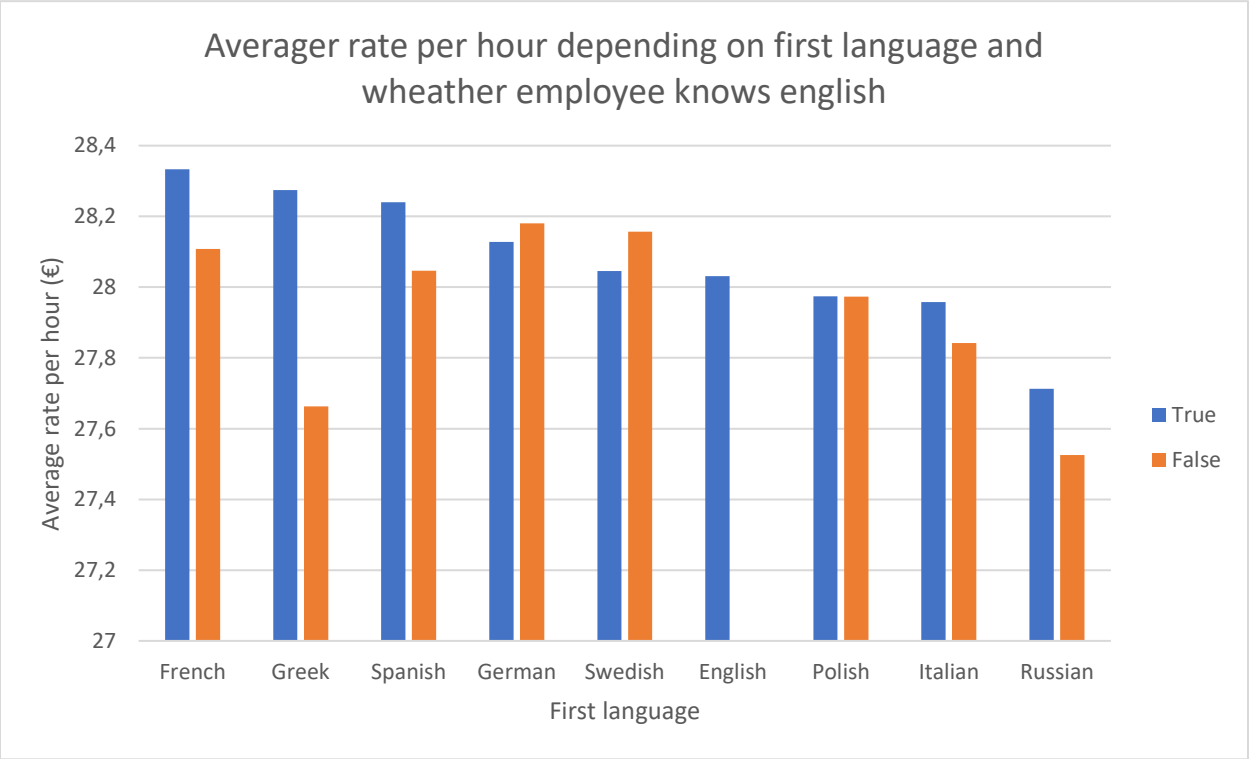
Wykres 17. Procent programistów pracujących w danym trybie

Jak widać w tym zawodzie bardzo duży udział pracowników pracuje w formie niestacjonarnej, jednak klasyczna forma stacjonarna wciąż jest najpopularniejsza.



Wykres 18. Średnia stawka za godzinę w euro w zależności od wielkości zespołu

W przypadku tego wykresu ciężko jest znaleźć konkretne zależności.



Wykres 19. Średnia stawka za godzinę w zależności od języka ojczystego i informacji czy pracownik zna angielski jako jeden z języków

Jak widać nie we wszystkich przypadkach (niemiecki, szwecki) osoby ze znajomością angielskiego nie zarabiają więcej, lecz w przypadku pozostałych języków różnice mogą być znaczniejsze. Co ciekawe również, osoby najwięcej zarabiające to osoby mówiące po francusku i grecku.

7.2. Podsumowanie - wnioski z analizy

8. Wnioski końcowe z realizacji projektu

8.1. Problemy

Tworzenie tabel:

Podstawowym problemem w tym przypadku było wprowadzenie odpowiednich ograniczeń – w przypadku wymiarów oznaczało to dodanie ograniczeń typu UNIQUE na wszystkie atrybuty z pominięciem klucza sztucznego – zabezpiecza to przed dodawaniem identycznych wymiarów. W analogiczny sposób zabezpieczona została unikatowość faktów.

Integration services:

Jednym z początkowych problemów który zauważyłam dopiero po pewnym czasie było rzutowanie typów bool – brak danych traktowany był jako False co powodowało przekłamanie danych. Z innymi

typami danych był analogiczny problem (np. z datą) przez co zdecydowałam się ostatecznie by większość danych sczytywać z pliku jako string i rzutować je na odpowiednie wartości dopiero w kolejnych krokach.

Tworzenie kostki:

Element przy tworzeniu kostki którego wcześniej nie używałam a zastosowałam w tym projekcie to atrybut Visible w Properties atrybutu – umożliwił on ukrycie kluczy wymiarów które w przypadku tej analizy jednie przeszkadzały. Poza tym o wiele większą uwagę zwróciłam na odpowiednie nazewnictwo atrybutów i hierarchii mając już wiedzę jak najwygodniej z tak przygotowanych wymiarów się potem korzysta.

Analiza danych:

W przypadku tej części problemów nie było, wybranie odpowiedniego typu wykresu do prezentowanych danych było w tym przypadku najtrudniejszym elementem, ale poza tym nie napotkałam się z żadnymi trudnościami.

8.2. Pozyskana wiedza i doświadczenie

Całościowo uważam, że o ile mogłabym mieć trudności w stworzeniu hurtowni dla większej ilości i bardziej złożonych źródeł danych, to jednocześnie jestem na tyle dobrze zapoznana z dostępnymi narzędziami (Tabulau, VS Integration Services, VS Multidimensional project) by wiedzieć gdzie zasięgnąć informacji w ramach napotkanych problemów.

W kwestii zapewnienia poprawności wczytywania plików, uważam że potrafię przeanalizować projekt dokładnie, lecz jednocześnie mam świadomość, że pewną wiedzę w temacie nabiera się poprzez doświadczenie i nie neguję, że mogę pozostawić pewne niedopatrzenia.

9. Źródła informacji użyte w etapie analizy danych

Jako, że tematyka związana była z zarobkami informatyków to nie mam do wskazania źródeł danych wykorzystanych w analizie.