

SVEUČILIŠTE JURJA DOBRILE U PULI

FAKULTET INFORMATIKE U PULI

Skladišta i rudarenje podataka

PRODAJA NEKRETNINA U MELBOURNE-U 2016 – 2018 GODINE

SEMINARSKI RAD

Mateo Kocev, 0303104813, izvanredni student

Informatika

Pula, 22.5.2024.

Contents

1. Uvod	3
2. Odabir dataseta / skupa podataka	4
2.1. Opis problema i cilj analize	4
2.2. Analiza podataka	5
3. Izrada relacijskog modela i baze podataka	7
3.1. Analiza i čišćenje skupa putem implementacije u Pythonu	7
3.2. Izrada konceptualnog modela	11
3.3. EER Diagram	13
3.4. Popunjavanje relacijskog modela	13
4. Izrada dimenzijskog modela	16
4.1. Diagram dimenzijskog modela	18
5. ETL Proces	19
5.1. Dimenzijska tablica 'Seller'	20
5.2. Dimenzijska tablica 'Sale_Info'	23
5.3. Dimenzijska tablica 'Property'	26
5.4. Dimenzijska tablica 'Location'	29
5.5. Dimenzijska tablica 'House_details'	32
5.6. Tablica činjenica 'Sale'	35
5.6.1. Prikaz tablice činjenica	37
6. Vizualizacija podataka	37
6.1. Prikaz najveće prodaje po regiji	38
6.2. Prosječna cijena nekretnina po regiji u periodu od 2016. do 2018.	39
6.3. Prikaz korištenja metoda prodaja	40

6.4. Prodajni trendovi podijeljeni po regiji	41
6.5. Odnos veličine zemljišta i cijene.....	42
6.6. Praćenje trenda prodaje po dostupnim komoditetima	43
6.7. Cijena nekretnine ovisno o dostupnim komoditetima	44
6.8. Prosječna cijena ovisno o vrsti nekretnine.....	45
6.9. Utjecaj godine gradnje na cijenu	46
6.10. Performansa prodavača.....	47
6.11. Performansa prodavača kroz vrijeme	48
7. Zaključak.....	49
8. Literatura.....	50

1. Uvod

U svijetu gdje se informacija smarta jednom od najvažnijih jedinica uspjeha, velike tvrtke ulažu velike količine kapitala kako bi prikupile i obradile velike količine informacija za što točniju poslovnu odluku te pomoću istih proširili utjecaj na tržištu.

Tržište nekretnina predstavlja jedan od najdinamičnijih i najvažnijih segmenata ekonomije, a razumijevanje njegovih trendova i promjena ključno je za donošenje informiranih poslovnih odluka. Melbourne, kao jedan od najbrže rastućih gradova u Australiji, nudi intrigantne prilike i izazove na tržištu nekretnina. Ovaj rad usmjeren je na analizu tržišta nekretnina u Melbourneu tijekom razdoblja od 2016. do 2018. godine, pružajući uvid u ključne faktore koji su utjecali na cijene i obujam prodaje.

Putem analize tržišta cilja dokazati da prikupljanjem i dobro obavljenom analizom podataka možemo doći do čiste informacije te dalje pokazati vrijednost integracije računala u poslovanje implementirajući prave alate.

2. Odabir dataseta / skupa podataka

Podaci su učitani iz CSV datoteke pronađene na poveznici:
<https://www.kaggle.com/datasets/ronikmalhotra/melbourne-housing-dataset>

U skupu podataka pronalazimo razne informacije o prodaji nekretnina u periodu od veljače 2016. godine do ožujka 2018. godine koje na lokaciji imaju sagrađenu i kuću.

Skup podataka smo podijelili na dva dijela kako bi pokušali realizirati simulaciju realistične situacije gdje nam se podaci neće nalaziti u jednom izvoru, već je na nama da dostupne podatke obradimo i pripremimo za rad.

2.1. Opis problema i cilj analize

Tržište nekretnina u Melbourneu suočava se s nizom izazova koji utječu na trgovanje nekretninama. Cijene nekretnina mogu biti vrlo promjenjive te stvoriti nesigurnost u donošenju odluka kod kupaca, prodajnih agenata i investitora što dovodi do financijskog gubitka ili propuštenih prilika. Ovaj problem možemo riješiti analizom povijesnih podataka te identifikacijom trendova na tržištu pružajući uvid ne samo u općenite trendove već gledajući performansu samih prodajnih agenata.

Problem nam također predstavlja veliku količinu faktora poput lokacije, veličine, dob nekretnine i dostupnih komoditeta koji mogu utjecati na cijene nekretnina. Teško je odrediti koji su najvažniji faktori koje moramo uzeti u obzir no kroz detaljnu analizu podataka ciljamo identificirati najveće utjecaje na kretanje tržišta kako bi mogli dovesti realne i konkurentne poslovne odluke.

Ciljamo optimizirati strategiju prodaje, kupnje i korištenja vremena kako bi minimizirali rizik te maksimizirali dobit financijskih sredstava donoseći pravilne poslovne odluke koristeći primjerene analitičke alate i modele. Također ciljamo održavati veću transparentnost i dostupnost informacija o često privatnim transakcijama kroz

prikupljanje, saniranje i obrađivanje podataka kako bi naši prodajni agenti i investitori mogli dovesti preciznu poslovnu odluku.

2.2. Analiza podataka

Skup podataka nam se sastoji o 22 atributa i ukupno 34,857 redaka gdje svaki individualni redak predstavlja zasebnu prodaju nekretnine. Skup podataka također sadrži prihvatljivu količinu kvalitativnih i kvantitativnih podataka među kojima nalazimo i veoma važnu vremensku dimenziju.

Atributi su sljedeći: Suburb, Address, Rooms, Type, Method, SellerG, Date, Distance, Postcode, Bedroom, Bathroom, Car, Landsize, BuildingArea, YearBuilt, CouncilArea, Latitude, Longitude, Regionname, Propertycount, ParkingArea, Price.

Po listi atributa primjećujemo nekoliko kategorija / vrsta atributa koje možemo podijeliti na sljedeći način:

- **Geografski atributi:** Suburb, Postcode, CouncilArea, Regionname, Latitude, Longitude predstavljaju neku lokaciju te opisuju razne aspekte lokacije u kojoj se nekretnina nalazi među kojima nalazimo regiju, kvart, poštanski broj i najvažnije koordinate koje nam daju preciznu informaciju o lokaciji.
- **Podaci o nekretnini:** Address, Landsize, buildingArea, YearBuilt, Propertycount, ParkingArea nam opisuju razne aspekte nekretnine koje bi potencijalni kupci uzimali u obzir za moguće modifikacije i buduće planove nadogradnje u reguliranom urbanom okolišu.
- **Podaci o kući:** Rooms, Type, Bedroom, Bathroom, Car nam opisuju detalje koji potencijalnim kupcima početno padaju na um te nam govore o dostupnim komoditetima kuće.

- **Podaci o prodavačima:** Skup podataka sadrži samo jedan atribut koji pripada prodavaču SellerG te predstavlja ime prodavača.
- **Podaci o prodaji:** Date, Method i Price opisuju informacije o prodaji nekretnine.

1	Suburb	Address	Rooms	Type	Method	SellerG	Date	Distance	Postcode	Bedroom	Bathroom	Car	Landsize	BuildingA	YearBuilt	CouncilAr	Latitude	Longitude	RegionName	PropertyType	ParkingAr	Price
2	Abbotsford	68 Studley	2	h	SS	Jellis	3/9/2016	2.5	3067	2	1	1	126	inf		Yarra City	-37.8014	144.9958	Northern I	4019	Carport	199300
3	Airport Wk	154 Halsey	3	t	PI	Nelson	3/9/2016	13.5	3042	3	2	1	303	225	2016	Moonee V	-37.718	144.878	Western M	3464	Detached	840000
4	Albert Park	105 Kerfer	2	h	S	hockingst	3/9/2016	3.3	3206	2	1	0	120	82	1900	Port Phillip	-37.8459	144.9555	Southern I	3280	Attached C	1275000
5	Albert Park	85 Richard	2	h	S	Thomson	3/9/2016	3.3	3206	2	1	0	159	inf		Port Phillip	-37.845	144.9538	Southern I	3280	Indoor	1455000

Slika 1: Prikaz atributa u CSV datoteci

Nakon analize svakog atributa možemo identificirati potrebne dimenzije za postizanje našeg cilja. Atributi koje ćemo koristiti primarno za analizu tržišta nekretnina glase:

- Latitude, Longitude, RegionName će nam dati prosječnu i preciznu informaciju o lokaciji te će nam pokazati moguće trendove vezane za lokaciju nekretnine.
- Karakteristike nekretnine i kuće poput Bedroom, Bathroom, Car i Landsize nam mogu ukazati trendove oko ponuđenih komoditeta koje potencijalni kupci traže te YearBuilt i Type nam ukazuje preference na vrstu kuće koja se cilja u prodajama
- Method i SellerG nam ukazuje trendove u načinu prodaje te nam pokazuje preferencije na tržištu.
- Date nam dodaje vremenski faktor gdje možemo ugledati promjenu trendova i performansi kroz vrijeme
- Price je najvažniji atribut u skupu s time da nam povezuje attribute i omogućuje spajanje atributa u svrhu usporedbe trendova

Iako neke attribute nećemo koristiti zbog nedostatka ili nezadovoljavajuće kvalitete podataka, skup nam pruža više nego dovoljno opširan skup podataka za izvođenje kvalitetne analize. Osim potencijalno malene količine iskoristivih redaka zbog nedostajućih podataka, naš skup zadovoljava ostale aspekte potrebne za izvršavanje projekta.

3. Izrada relacijskog modela i baze podataka

Kako bi stvorili bazu podataka i relacijski model za pripremu analize tržišta nekretnina, prvo ćemo primijeniti alate dostupne u programskom jeziku Python kako bi analizirali stanje integriteta podataka u našoj CSV datoteci. Sa Python programskim jezikom imamo dostupnu biblioteku *pandas*, koja nam omogućava rad na našim podacima putem manipulacije *pandas* specifičnog objekta *DataFrame* u koji učitavamo naše podatke te putem kojeg možemo jednostavno izvesti analizu, čišćenje i modifikaciju podataka u našoj CSV datoteci.

Nakon analize skupa koristeći Python alate, prelazimo na modeliranje i implementaciju baze podataka koristeći *sqlalchemy* i *MySQL Workbench*. *sqlalchemy* je biblioteka dostupna u Pythonu koja omogućava deklarativno mapiranje modela, tj. definiranje modela baze podataka koristeći Python klase, omogućava upravljanje sesijama za interakciju s bazom podataka gdje možemo implementirati promjene i upravljati relacijama između modela. U ovoj situaciji ćemo primijeniti *sqlalchemy* kako bi implementirali naš relacijski model u MySQL bazu podataka učitavajući podatke iz CSV-a direktno u MySQL bazu.

3.1. Analiza i čišćenje skupa putem implementacije u Pythonu

Implementacijom sljedeće Python skripte smo analizirali, očistili te raspodijelili skup podataka na dva dijela kako bi simulirali realnu situaciju gdje nam podaci dolaze sa više izvora.

Korišteni alati:

- Poveznica na službenu stranicu Python dokumentacije: docs.python.org/3/
- Poveznica na službenu stranicu *pandas* dokumentacije: pandas.pydata.org/docs/


```

1. import pandas as pd
2.
3. CSV_FILE_PATH = "Melbourne_housing.csv"
4. df = pd.read_csv(CSV_FILE_PATH, delimiter=',')
5.
6. print("Veličina skupa (broj redaka, broj stupaca):", df.shape)
7.
8. print("Nazivi stupaca:", df.columns.tolist())
9.
10. for column in df.columns:
11.     print(f"Jedinstvene vrijednosti u stupcu {column}: {df[column].nunique()}")
12.
13. print("Tipovi podataka po stupcu:\n", df.dtypes)
14.
15. print("Broj nedostajućih vrijednosti po stupcu:\n", df.isna().sum())
16.
17. print("CSV size before: ", df.shape)
18.
19. df = df.dropna()
20. print("CSV size after: ", df.shape)
21. print(df.head())
22.
23. df20 = df.sample(frac=0.2, random_state=1)
24. df = df.drop(df20.index)
25. print("CSV size 80: ", df.shape)
26. print("CSV size 20: ", df20.shape)
27.
28. df.to_csv("Melbourne_housing_PROCESSED.csv", index=False)
29. df20.to_csv("Melbourne_housing_PROCESSED_20.csv", index=False)

```

U skripti pozivamo biblioteku pandas te učitavamo CSV datoteku sa lokalnog foldera (skripta se mora nalaziti u istom folderu da bi korištena ruta radila) u pandas DataFrame objekt.

Putem `.shape` metode provjeravamo veličinu učitano DataFrame-a

```
Veličina skupa (broj redaka, broj stupaca): (34857, 22)
```

Slika 2: Ispis veličine DF-a u konzoli

Koristeći `.tolist()` metodu na kolonama DF-a smo ispisali sve učitane atribute

```
Nazivi stupaca: ['Suburb', 'Address', 'Rooms', 'Type', 'Method', 'SellerG', 'Date', 'Distance', 'Postcode', 'Bedroom', 'Bathroom', 'Car', 'Landsize', 'BuildingArea', 'YearBuilt', 'CouncilArea', 'Latitude', 'Longitude', 'Regionname', 'Propertycount', 'ParkingArea', 'Price']
```

Slika 3: Ispis atributa u konzoli

Iterirajući kroz kolone DF-a ispisujemo količinu jedinstvenih vrijednosti u skupu podataka. Koristeći `.nunique()` dobijamo količinu jedinstvenih vrijednosti na osi.

```
Jedinstvene vrijednosti u stupcu Suburb: 351
Jedinstvene vrijednosti u stupcu Address: 34009
Jedinstvene vrijednosti u stupcu Rooms: 12
Jedinstvene vrijednosti u stupcu Type: 3
Jedinstvene vrijednosti u stupcu Method: 9
Jedinstvene vrijednosti u stupcu SellerG: 388
Jedinstvene vrijednosti u stupcu Date: 78
Jedinstvene vrijednosti u stupcu Distance: 215
Jedinstvene vrijednosti u stupcu Postcode: 211
Jedinstvene vrijednosti u stupcu Bedroom: 15
Jedinstvene vrijednosti u stupcu Bathroom: 11
Jedinstvene vrijednosti u stupcu Car: 15
Jedinstvene vrijednosti u stupcu Landsize: 1684
Jedinstvene vrijednosti u stupcu BuildingArea: 994
Jedinstvene vrijednosti u stupcu YearBuilt: 160
Jedinstvene vrijednosti u stupcu CouncilArea: 33
Jedinstvene vrijednosti u stupcu Latitude: 13402
Jedinstvene vrijednosti u stupcu Longitude: 14524
Jedinstvene vrijednosti u stupcu Regionname: 8
Jedinstvene vrijednosti u stupcu Propertycount: 342
Jedinstvene vrijednosti u stupcu ParkingArea: 8
Jedinstvene vrijednosti u stupcu Price: 2871
```

Slika 4: Ispis količine jedinstvenih vrijednosti po atributu

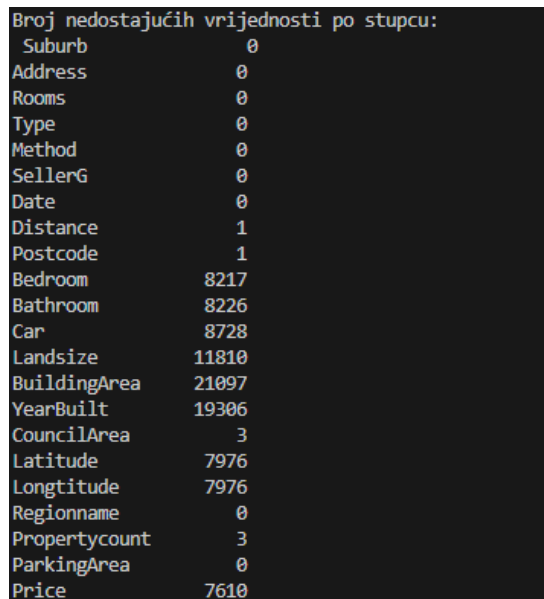
Koristeći `.dtypes` metodu provjeravamo vrstu podataka koje nalazimo među atributima

```
Tipovi podataka po stupcu:
Suburb      object
Address     object
Rooms       int64
Type        object
Method      object
SellerG     object
Date        object
Distance    float64
Postcode    float64
Bedroom     float64
Bathroom    float64
Car         float64
Landsize    float64
BuildingArea object
YearBuilt   float64
CouncilArea object
Latitude    float64
Longitude   float64
Regionname  object
Propertycount float64
ParkingArea object
Price       float64
```

Slika 5: Prikaz vrsta podataka u DF-u

Date je zapisan kao object umjesto podržane jedinice vremena u pandas DF-u no to se lagano ispravi prije ubacivanja u bazu podataka.

U zadnjem koraku analize provjeravamo količinu nedostajućih vrijednosti po stupcu koristeći `.isna()` metodu te sumirajući vrijednosti po atributima.



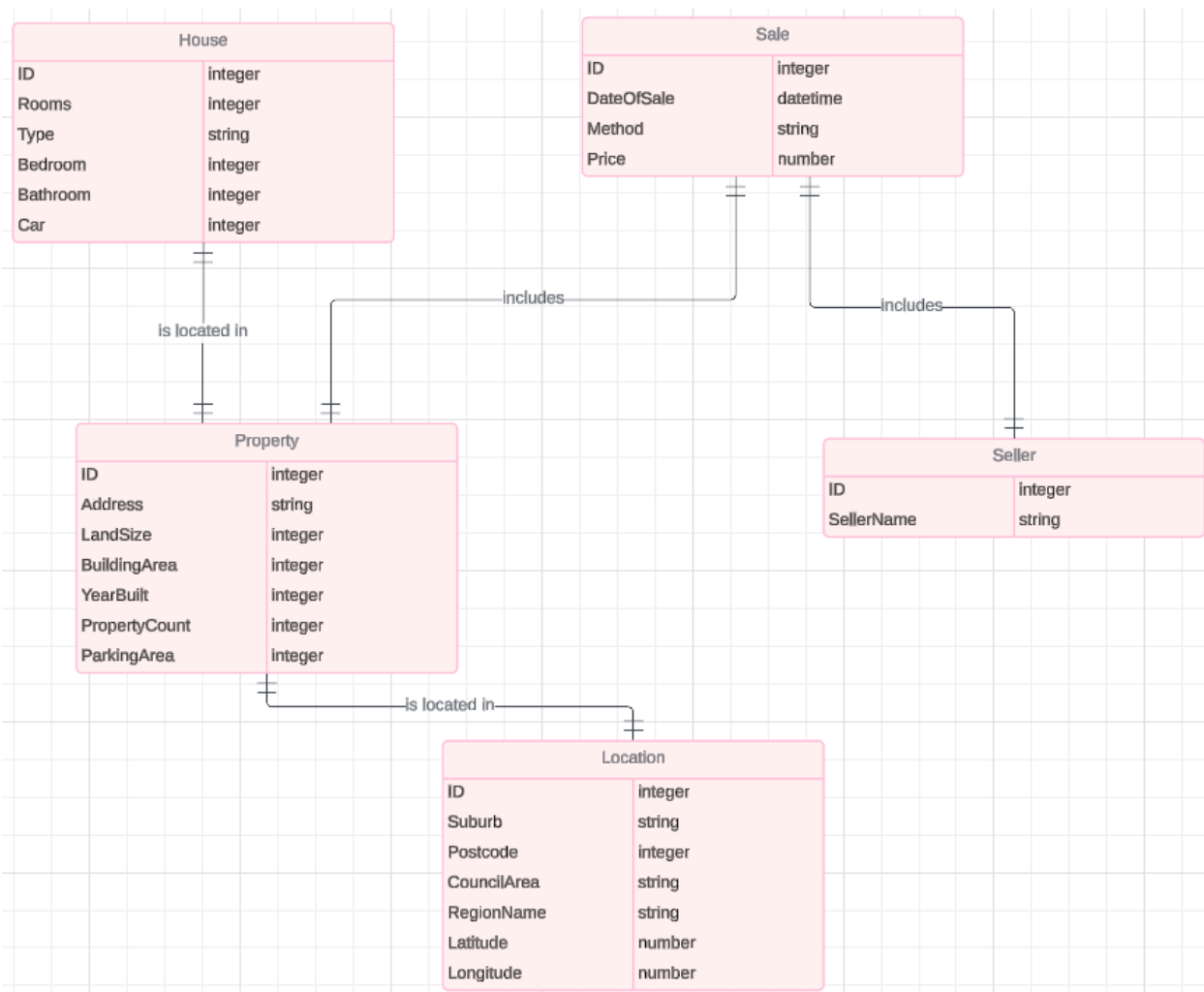
```
Broj nedostajućih vrijednosti po stupcu:
Suburb          0
Address         0
Rooms          0
Type           0
Method         0
SellerG        0
Date           0
Distance       1
Postcode       1
Bedroom       8217
Bathroom      8226
Car           8728
Landsize      11810
BuildingArea  21097
YearBuilt     19306
CouncilArea    3
Latitude      7976
Longitude     7976
Regionname     0
Propertycount  3
ParkingArea    0
Price         7610
```

Slika 6: Prikaz količine nedostajećih podataka po atributu

U sljedećem koraku provjeravamo početnu veličinu DF-a te brišemo sve retke sa null vrijednostima. Tijekom provjere preostalih podataka uočavamo da nam je ostalo oko 9000 preciznih podataka o prodajama. Iako smo izgubili veliki dio podataka, čisti i ispunjeni skup podataka će nam pružati preciznije rezultate tijekom analize.

Za kraj nasumično dijelimo skup podataka na 80% i 20% te ih spremamo u zasebne CSV datoteke koje se nalaze na lokalnoj ruti.

3.2. Izrada konceptualnog modela



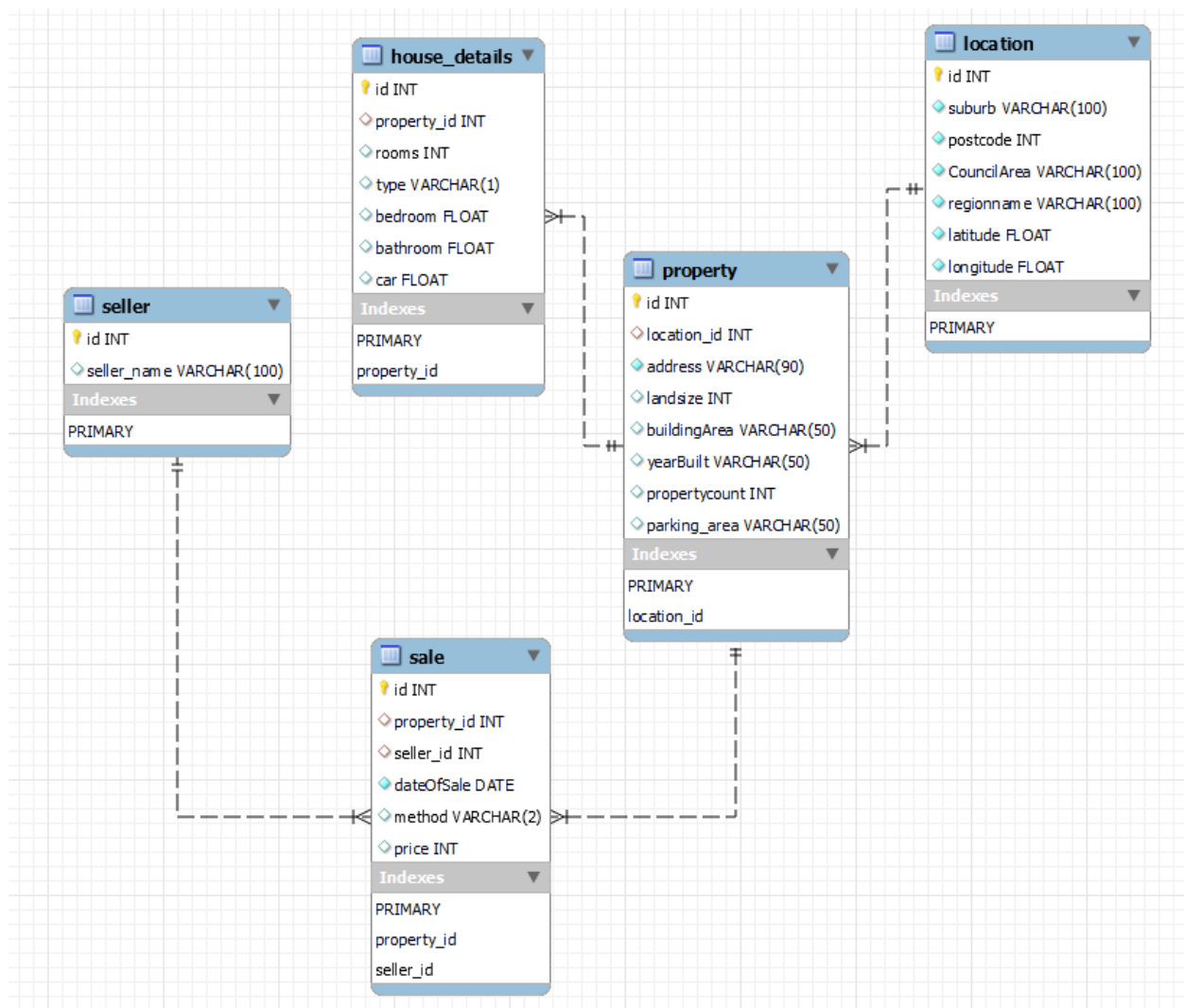
Slika 7: Konceptualni model

Entiteti konceptualnog modela:

- **SALE:** prodaja je glavni entitet te sadrži attribute id, DateOfSale(datum prodaje), Method(metoda prodaje) i Price(cijena). Opisuje vrijeme, cijenu i metodu prodaje nekretnine. Metoda prodaje može biti S(prodano na dražbi), SP(prodano prije dražbe), PI(prodano van dražbe), VB(prodano agentu u pokušaju da digne cijenu), SA(prodano poslije dražbe)

- **SELLER:** prodavač je drugi entitet. Sadrži atribut id, SellerName(ime prodajnog agenta) te se povezuje na prodavača pošto opisuje koji prodavač je povezan sa kojom prodajom. Jedan prodavač može biti povezan na više prodaja.
- **PROPERTY:** entitet koji opisuje osnovne informacije o nekretnini te se povezuje na informacije o prodaji. Sastoji se od id, Address(adresa), LandSize(veličina zemljišta), BuildingArea(dopuštena veličina gradnje), YearBuilt(godina gradnje), PropertyCount(broj nekretnina uključenih u zoni) i ParkingArea(vrsta dostupnog parkinga). Parking area ima 8 mogućih stanja te može biti Attached Garage(garaža spojena na kuću), Detached Garage(garaža odvojena od kuće), Carport(otvoreni parking sa zaštitom), Indoor(zatvoreni javni parking), Parkade(javni parking na više katova), Underground(podzemni parking), Outdoor Stall(javni otvoreni parking), Parking Pad(privatno mjesto pred kućom).
- **HOUSE DETAILS:** entitet koji se veže na nekretninu, opisuje vrstu i komoditete dostupne kući. Sastoji se od atributa id, Rooms(broj prostorija), Type(vrsta kuće), Bedroom(broj spavaćih soba), Bathroom(broj kupaoana), Car(kapacitet vozila). Type ima 3 moguća stanja te može biti h(Individualna kuća), t(gradska kuća), u (stan)
- **LOCATION:** entitet koji opisuje informacije o preciznoj lokaciji nekretnine, nadovezuje se na nekretninu. Sastoji se od Suburb(naselje), Postcode(poštanski broj), CouncilArea(općinsko područje), RegionName(regija), Latitude(zemljopisna širina), Longitude(zemljopisna dužina). Koordinate su provjerene te se mogu koristiti u prikazu heatmape koja ukazuje na trendove.

3.3. EER Diagram



Slika 8: Extended Entity-Relationship Diagram

3.4. Popunjavanje relacijskog modela

Popunjavanje relacijskog modela je obavljeno koristeći skriptu napravljenu u programskom jeziku Python implementirajući biblioteku *sqlalchemy* kako bi uspostavili vezu sa MySQL bazom podataka.

```

1. import pandas as pd
2. from sqlalchemy import create_engine, Column, Integer, String, Float, ForeignKey, Date, UniqueConstraint
3. from sqlalchemy.orm import sessionmaker, relationship, declarative_base
4. from datetime import datetime
5.
6. CSV_FILE_PATH = "Melbourne_housing_PROCESSED.csv"
7. df = pd.read_csv(CSV_FILE_PATH, delimiter=',')
8.
9. Base = declarative_base()
10.
11. class Location(Base):
12.     __tablename__ = 'location'
13.     id = Column(Integer, primary_key=True)
14.     suburb = Column(String(100), nullable=False)
15.     postcode = Column(Integer, nullable=False)
16.     CouncilArea = Column(String(100), nullable=False)
17.     regionname = Column(String(100), nullable=False)
18.     latitude = Column(Float, nullable=False)
19.     longitude = Column(Float, nullable=False)
20.
21. class Property(Base):
22.     __tablename__ = 'property'
23.     id = Column(Integer, primary_key=True)
24.     location_id = Column(Integer, ForeignKey('location.id'))
25.     address = Column(String(90), nullable=False)
26.     landsize = Column(Integer)
27.     buildingArea = Column(String(50))
28.     yearBuilt = Column(String(50))
29.     propertycount = Column(Integer)
30.     parking_area = Column(String(50))
31.
32. class HouseDetails(Base):
33.     __tablename__ = 'house_details'
34.     id = Column(Integer, primary_key=True)
35.     property_id = Column(Integer, ForeignKey('property.id'))
36.     rooms = Column(Integer)
37.     type = Column(String(1))
38.     bedroom = Column(Float)
39.     bathroom = Column(Float)
40.     car = Column(Float)
41.
42. class Seller(Base):
43.     __tablename__ = 'seller'
44.     id = Column(Integer, primary_key=True)
45.     seller_name = Column(String(100))
46.
47. class Sale(Base):
48.     __tablename__ = 'sale'
49.     id = Column(Integer, primary_key=True)
50.     property_id = Column(Integer, ForeignKey('property.id'))
51.     seller_id = Column(Integer, ForeignKey('seller.id'))
52.     dateOfSale = Column(Date, nullable=False)
53.     method = Column(String(2))
54.     price = Column(Integer)
55.
56. engine = create_engine('mysql+pymysql://root:root@localhost:3306/melhouse', echo=False)
57.
58. Base.metadata.create_all(engine)
59.
60. Session = sessionmaker(bind=engine)
61. session = Session()
62.

```

```

63. def convert_date(date_str):
64.     return datetime.strptime(date_str, '%d/%m/%Y')
65.
66. for index, row in df.iterrows():
67.
68.     location = Location(suburb=row['Suburb'],
69.                         postcode=row['Postcode'],
70.                         CouncilArea=row['CouncilArea'],
71.                         regionname=row['Regionname'],
72.                         latitude=row['Latitude'],
73.                         longitude=row['Longitude'])
74.     session.add(location)
75.     session.flush()
76.
77.     property = Property(address=row['Address'],
78.                         landsize=row['Landsize'],
79.                         buildingArea=row['BuildingArea'],
80.                         yearBuilt=row['YearBuilt'],
81.                         propertycount=row['Propertycount'],
82.                         parking_area=row['ParkingArea'])
83.     property.location_id = location.id
84.     session.add(property)
85.     session.flush()
86.
87.     house_details = HouseDetails(rooms=row['Rooms'],
88.                                   type=row['Type'],
89.                                   bedroom=row['Bedroom'],
90.                                   bathroom=row['Bathroom'],
91.                                   car=row['Car'])
92.     house_details.property_id = property.id
93.     session.add(house_details)
94.     session.flush()
95.
96.     seller = Seller(seller_name=row['SellerG'])
97.     session.add(seller)
98.     session.flush()
99.
100.    sale = Sale(dateOfSale=convert_date(row['Date']),
101.               method=row['Method'],
102.               price=row['Price']
103.             )
104.    sale.property_id = property.id
105.    sale.seller_id = seller.id
106.    session.add(sale)
107.    session.flush()
108.
109. session.commit()
110. session.close()
111. print("Tablice su popunjene.")

```

Program učitacva podatke iz CSV datoteke pozvanj na definiranoj ruti i pohranjuje ih u bazu podataka. Koristeći sqlalchemy definirane su tablice location, property, house_details, seller i sale. Podaci se iteriraju redak po redak, gdje se za svaki redak popunjavaju objekti tablica i spremaju u bazu podataka. Na kraju se svi podaci trajno pohranjuju i veza sa bazom se zatvara.

4. Izrada dimenzijskog modela

Izrada dimenzijskog modela služi za organizaciju podataka radi poboljšanja performance i jednostavnosti analize. Svrha dimenzijskog modela je olakšati i omogućiti učinkovitu pohranu, pristup, modifikaciju i analizu velikih količina podataka, koji često dolaze iz različitih izvora. Dimenzijski model se sastoji od jedne činjenice i pet različitih dimenzija koje ćemo primijeniti kako bi mjerljivom podatku predstavili kontekst, tj. dimenziju.

Dimenzijski model je također optimiziran za brzu obradu podataka zbog implementirane strukture zvijezde koja omogućuje brzo pridruživanje tablica i smanjuje potrebno vrijeme za obrađivanje zahtjeva, što nam dalje olakšava složene analitičke upite, među kojima nalazimo agregaciju, filtriranje i grupiranje podataka.

Među alternativama nalazimo i strukturu pahuljice, no za veličinu našeg skupa podataka, nije potrebna te granana struktura modela bi patila na performansama operacija na podacima.

Sam model implementiran u projektu se sastoji od tablice činjenica *sale* i pet dimenzijskih tablica koje glase: *sale_info*, *seller*, *property*, *location* i *house_details*.

Tablica činjenica sastoji se od jedne mjere – cijena prodaje nekretnine. Cijena prodaje služi kao osnovni indikator informacije o prodaji nekretnine te se na nju nadovezuju ostale ključne dimenzije kroz strane ključeve za prodavača, prodaju, nekretninu, kuću i lokaciju.

Dimenzijska tablica *location* sadrži hijerarhiju podataka o lokaciji, od samog naselja ('suburb'), poštanskog broja ('postcode'), općinskog područja ('CouncilArea'), do regije ('regionname') u kojoj se nalazi nekretnina. Također nalazimo i geografske koordinate ('latitude' i 'longitude') koje omogućuju prostornu analizu podataka. Tablica sadrži surogat ključ ('id') koji služi kao primarni ključ tablice.

Dimenzijska tablica *sale_info* sadrži podatke o datumu prodaje ('dateOfSale') i metodi prodaje ('method') koji omogućavaju analizu podataka po danu, mjesecu i godini.

Vremenska dimenzija nije sporo mijenjajuća te se ne mijenja nakon unosa. Također sadrži surogat ključ ('id') koji služi kao primarni ključ tablice.

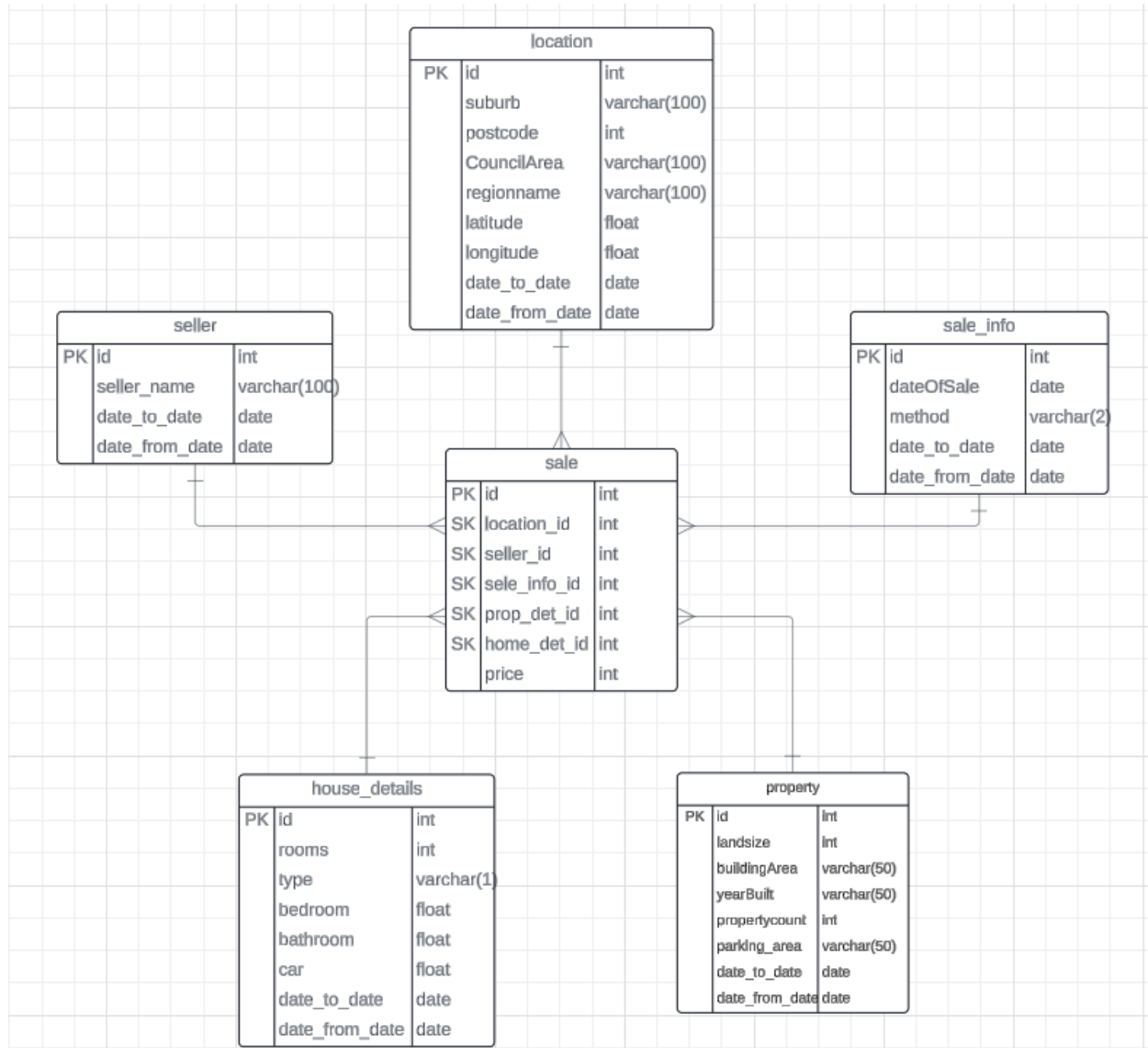
Dimenzijska tablica *property* sadrži podatke o godini izgradnje ('yearBuilt'), veličine nekretnine ('landsize') i površine dopuštene gradnje ('buildingArea'). Dimenzija omogućava grupiranje prema godini izgradnje i analizu utjecaja veličine parcele na činjenicu. Također sadrži surogat ključ ('id') koji služi kao primarni ključ tablice.

Dimenzijska tablica *house_details* sadrži detalje o kući te u njoj nalazimo broj prostorija ('rooms'), vrsta kuće ('type'), broj spavaćih soba ('bedroom'), broj kupaoonica ('bathroom') i kapacitet parkirnog mjesta ('car'). Omogućava filtriranje i analizu po komoditetima dostupnim sa nekretninom. Također sadrži surogat ključ ('id') koji služi kao primarni ključ tablice.

Dimenzijska tablica *seller* sadrži jedan podatak – ime prodajnog agenta. Omogućava nam da pratimo performansu i trendove među uspješnim agentima. Također sadrži surogat ključ ('id') koji služi kao primarni ključ tablice.

Pružajući ukupni pogled na dimenzijski model može se uočiti da su sve dimenzije osim vremenske sporo mijenjajuće te se njihove promjene mogu pratiti kroz attribute ('version', 'date_from_date' i 'date_to_date'). Dostupna je odlična osnova za analizu podataka o prodaji nekretnina prateći utjecaje dimenzija na cijenu prodaje što će dovesti do bolje informiranih poslovnih odluka i otvoriti put za bolje strateško planiranje.

4.1. Diagram dimenzijskog modela



Slika 9: Dimenzijski model

5. ETL Proces

ETL (*extract, transform, load*) je process koji se koristi za prikupljanje podataka iz potencijalno različitih izvora, transformaciju i pripremu te spremanje u ciljano skladište podataka. Sastoji se od tri glavna koraka:

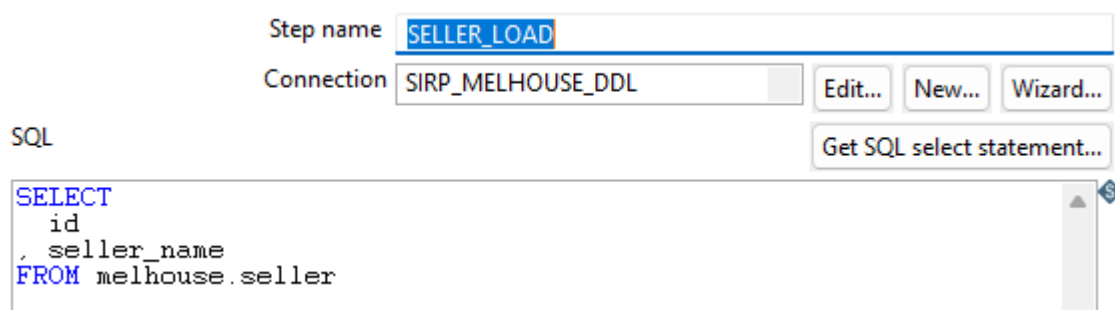
1. **Extract** (Izvlačenje): najčešće se podaci prikupljaju sa više izvora poput relacijskih baza, CSV datoteka, API-eva te ostale metode skladištenja podataka. Cilj je prikupiti što više relevantnih podataka za obradu.
2. **Transform** (Transformacija): sljedeći korak uključuje pripremu podataka za skladištenje podataka je priprema, tj. transformacija istih u format koji će omogućiti učitavanje istih bez problema u ciljni sustav. Transformacija može uključivati čišćenje, agregaciju, normalizaciju, filtriranje i sortiranje podataka.
3. **Load** (Učitavanje): posljednji korak ETL-a je učitavanje podataka u namijenjeno skladište podataka. Cilj je osigurati da su svi relevantni podaci dostupni za rad.

Za trenutni korak korišten je *Pentaho Data Integration tool* koji pomoću jednostavnog sučelja omogućava izradu ETL procesa za svaku dimenzijsku tablicu.

Svaka tablica će se sastojati o procesa spajanja podataka iz dva različita izvora gdje 80% posto podataka će doći od relacijskog modela spremljenog u relacijskoj bazi podataka a preostalih 20% dolaze iz CSV datoteke. Nakon učitavanja preoblikujemo strukturu podataka za stapanje i spremanje u dimenzijski model koji se također skladišti u MySQL bazu podataka.

5.1. Dimenzijska tablica 'Seller'

1. Dohvat podataka iz relacijske baze: Putem opcije 'Table input' učitavamo podatke iz relacijske baze podataka te ih sortiramo i osiguravamo da su svi podaci točnog formata i vrste u 'Select Values'. Konvertiramo string iz Binary u Normal.



Step name: **SELLER_LOAD**

Connection: **SIRP_MELHOUSE_DDL** [Edit...] [New...] [Wizard...]

SQL: `SELECT
 id
 , seller_name
FROM melhouse.seller`

[Get SQL select statement...]

Slika 10: Table Input od Seller tablice

2. Dohvat podataka iz CSV datoteke: Putem CSV file input učitavamo podatke iz CSV datoteke. Sa prvim 'Select values' selektiramo, preimenujemo i osiguravamo vrstu i format podataka povučenih iz CSV-a. Sa 'Add sequence' elementom generiramo atribut 'id' od zadnjeg poznatog id-a u relacijskoj bazi te u drugom 'Select values' osiguravamo točan redoslijed atributa za spajanje podataka. Redoslijed glasi: 'id' (Integer), 'seller_name' ('String').


Fields to alter the meta-data for :


#	Fieldname	Rename to	Type	Length	Precision	Binary to N
1	sellerG	seller_name	String			Y


Slika 11: Formatiranje podataka

Use counter to calculate sequence? ☒

Counter name (optional)

Start at value 

Increment by 

Maximum value 

Slika 12: Generiranje id za tablicu Seller

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	id			
2	seller_name			

Slika 13: Uređivanje redoslijeda atributa u tablici Seller

3. Merge i spremanje: Uz zadnjem koraku se spajaju dvije tablice po atributu 'id' i sortiraju se po 'Ascending' te se podatke sprema postavljajući ključ 'id' i Fields se popunjava preostalim podacima, u ovom slučaju 'seller_name'. Tehnički ključ 'seller_tk' je generiran pomoću ugrađene funkcije u 'Dimension lookup/update'. Za kraj koristimo ugrađenu SQL funkcionalnost da bi tablicu stvorili u novoj bazi podataka za dimenzijski model i eliminiramo u SQL generirani *null* red.

Sorted merge

Step name

Fields :

#	Fieldname	Ascending
1	id	Y

Slika 14: Sorted merge na Seller tablici

Dimension lookup/update

Step name: **SELLER_DIM_CREATE**

Update the dimension? ☒

Connection: **SIRP_MELHOUSE_DDL** [Edit...] [New...] [Wizard...]

Target schema: **MELHOUSE_DDL** [Browse...]

Target table: **SELLER_DIM** [Browse...]

Commit size: **100**

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all): **5000**

Keys | Fields

Lookup/Update fields

#	Dimension field	Stream field to compare with	Type of dimension update
1	seller_name	seller_name	Insert

Technical key field: **seller_tk** [New name]

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field: **version**

Stream Datefield:

Date range start field: **date_from** Min. year: **1900**

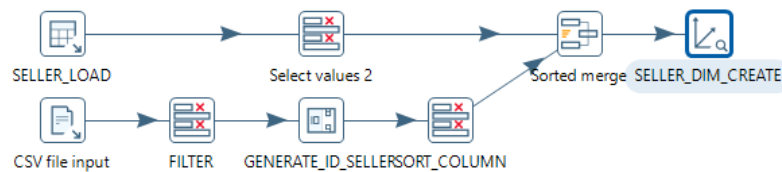
Use an alternative start date? ☐ <Select Option>

Table date range end: **date_to** Max. year: **2199**

[OK] [Cancel] [Get Fields] [SQL]

[Help]

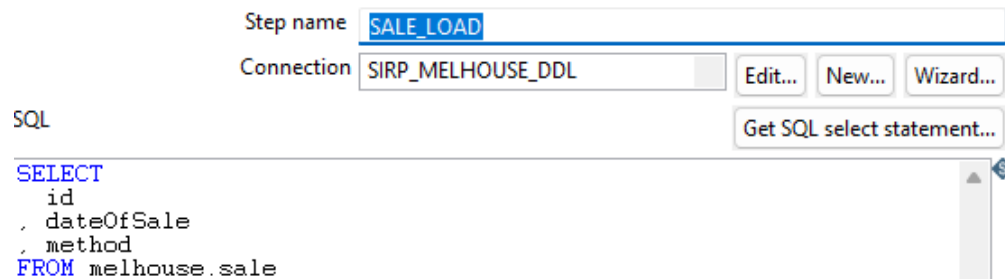
Slika 15: Dimension update za Seller tablicu



Slika 16: ETL proces za Seller tablicu

5.2. Dimenzijska tablica 'Sale_Info'

1. Dohvat podataka iz relacijske baze: Koristeći 'Table input' ponovno se učitavaju podaci iz relacijske baze podataka izostavljajući cijenu te ih se sortira i osigurava da su svi podaci točnog formata i vrste u 'Select Values'.



Step name: SALE_LOAD

Connection: SIRP_MELHOUSE_DDL

SQL: SELECT id, dateOfSale, method FROM melhouse.sale

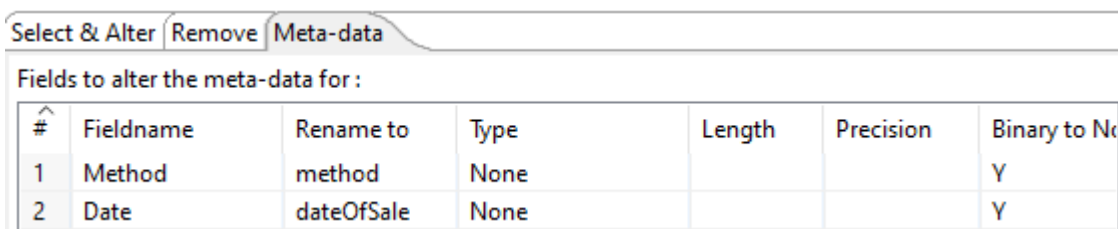
Slika 17: Učitavanje podataka za 'sale_info' iz MySQL

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to Norm
1	dateOfSale		Date			Y
2	method		None	2		Y

Slika 18: Modifikacija podataka za 'sale_info' tablicu

2. Dohvat podataka iz CSV datoteke: Putem 'CSV file input' se uzimaju podaci iz CSV datoteke. Sa prvim 'Select values' se obavlja selekcija podataka te se preimenuje i osigura vrsta i format podataka povučenih iz CSV-a. Sa 'Add sequence' elementom se generira atribut 'id' od zadnjeg poznatog id-a u relacijskoj bazi te u drugom 'Select values' se osigurava točan redoslijed atributa za spajanje podataka. Redoslijed glasi: 'id' (Integer), 'dateOfSale' (Date), 'method' (String).



Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to Norm
1	Method	method	None			Y
2	Date	dateOfSale	None			Y

Slika 19: Formatiranje podataka 'Sale_info' tablice

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional)

Start at value

Increment by

Maximum value

Slika 20: Generiranje atributa 'id' za tablicu 'sale_info'

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	id			
2	dateOfSale			
3	method			

Slika 21: Sortiranje redoslijeda podataka (priprema za merge)

- Merge i spremanje: Uz zadnjem koraku se spajaju dvije tablice po atributu 'id' i sortiraju se po 'Ascending' te se podatke sprema postavljajući ključ 'id' i Fields se popunjava preostalim podacima – dateOfSale, method. Generira se tablica pomoću SQL funkcije te se podaci spremaju. U MySQL-u se briše *null* red.

Sorted merge

Step name

Fields :

#	Fieldname	Ascending
1	id	Y

Slika 22: Merge po 'id' atributu

Dimension lookup/update

Step name: SALE_INFO_DIM_CREATE

Update the dimension? ☒

Connection: SIRP_MELHOUSE_DDL Edit... New... Wizard...

Target schema: MELHOUSE_DDL Browse...

Target table: SALE_INFO_DIM Browse...

Commit size: 100

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all): 5000

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	id	id

Technical key field: sale_info_tk New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field: version

Stream Datefield:

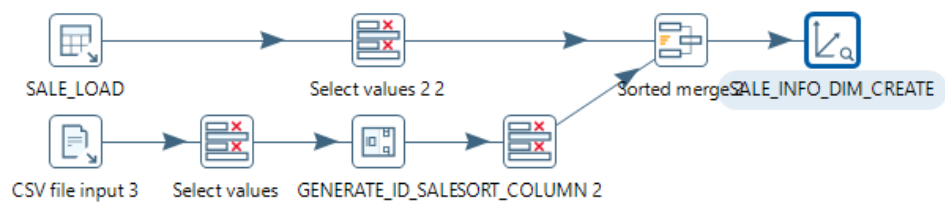
Date range start field: date_from Min. year: 1900

Use an alternative start date? ☐ <Select Option>

Table date range end: date_to Max. year: 2199

OK Cancel Get Fields SQL

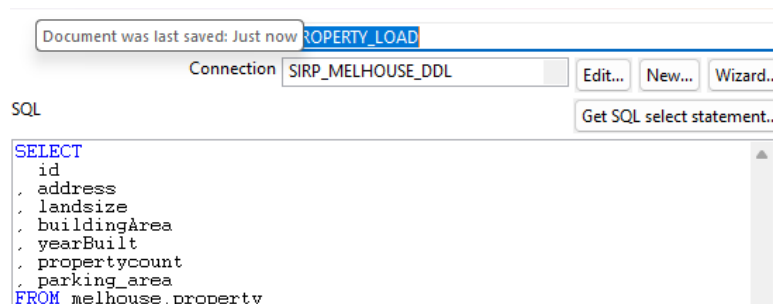
Slika 23: Definicija dimenzijske tablice 'sale_info'



Slika 24: ETL proces za 'sale_info'

5.3. Dimenzijska tablica 'Property'

1. Dohvat podataka iz relacijske baze: Koristeći 'Table input' ponovno se učitavaju podaci iz relacijske baze podataka te ih se sortira i osigurava da su svi podaci točnog formata i vrste u 'Select Values'. Osiguravaju se tipovi podatak 'landsize' na Integer dužine 9, 'buildingArea' na String dužine 50, 'yearBuilt' na String dužine 50, 'propertycount' na Integer dužine 9 i 'parking_area' na String dužine 50. Svaki podatak kovertiramo iz binarnog formata u normalni kako bi izbjegli probleme kod merge-a.



Slika 25: Učitavanje podatak iz MySQL za 'Property' tablicu

The screenshot shows a software window with a title bar that says "Select & Alter" and a tab labeled "Meta-data". Below the title bar is a label "Fields to alter the meta-data for:". Below this is a table with 7 rows and 7 columns. The columns are: #, Fieldname, Rename to, Type, Length, Precision, and Binary to N. The rows are:

#	Fieldname	Rename to	Type	Length	Precision	Binary to N
1	id		None	9	0	N
2	address		None	90		N
3	landsize		Integer	9	0	Y
4	buildingArea		String	50		Y
5	yearBuilt		String	50		Y
6	propertycount		Integer	9	0	Y
7	parking_area		String	50		Y

Slika 26: Formatiranje podataka iz MySQL

2. Dohvat podataka iz CSV datoteke: Putem 'CSV file input' se uzimaju podaci iz CSV datoteke. Sa prvim 'Select values' se obavlja selekcija podataka te se preimenuje i osigura vrsta i format podataka povučenih iz CSV-a. Sa 'Add sequence' elementom se generira atribut 'id' od zadnjeg poznatog id-a u relacijskoj bazi te u drugom 'Select values' se osigurava točan redoslijed atributa za spajanje

podataka. Redoslijed glasi: 'id' (Integer), 'Address' (String), 'landsize' (Integer), 'buildingArea' (String), 'yearBuilt' (String), 'propertycount' (Integer) i 'parking_area' (String).

Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to
1	Address	address	None	90		Y
2	Landsize	landsize	Integer	9	0	Y
3	BuildingArea	buildingArea	String	50		Y
4	YearBuilt	yearBuilt	String	50		Y
5	Propertycount	propertycount	Integer	9	0	Y
6	ParkingArea	parking_area	String	50		Y

Get fields to change

Slika 27: Formatiranje za CSV podatke

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional)

Start at value 7113

Increment by 1

Maximum value 999999999

Slika 28: Generiranje 'id' atributa

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	id			
2	address			
3	landsize			
4	buildingArea			
5	yearBuilt			
6	propertycount			
7	parking_area			

Get fields to select

Edit Mapping

Slika 29: Sortiranje redoslijeda atributa

3. Merge i spremanje: Uz zadnjem koraku se spajaju dvije tablice po atributu 'id' i sortiraju se po 'Ascending' te se podatke sprema postavljajući ključ 'id' i Fields se popunjava preostalim podacima – id, address, landsize, buildingArea, yearBuilt, propertycount i parking_area. Generira se tablica pomoću SQL funkcije te se podaci spremaju. U MySQL-u se briše *null* red.

Step name **Sorted merge**

Fields :

#	Fieldname	Ascending
1	id	Y

Slika 30: Merge po atributu 'id'

Step name **PROPERTY_DIM_CREATE**

Update the dimension? ☒

Connection **SIRP_MELHOUSE_DDL** Edit... New... Wizard...

Target schema **MELHOUSE_DDL** Browse...

Target table **PROPERTY_DIM** Browse...

Commit size **100**

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all) **5000**

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	id	id

Technical key field **property_tk** New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field **version**

Stream Datefield

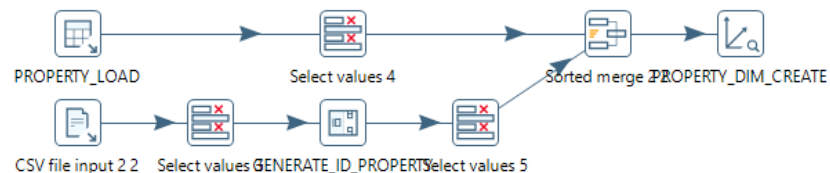
Date range start field **date_from** Min. year **1900**

Use an alternative start date? ☐ <Select Option>

Table date range end **date_to** Max. year **2199**

OK Cancel Get Fields SQL

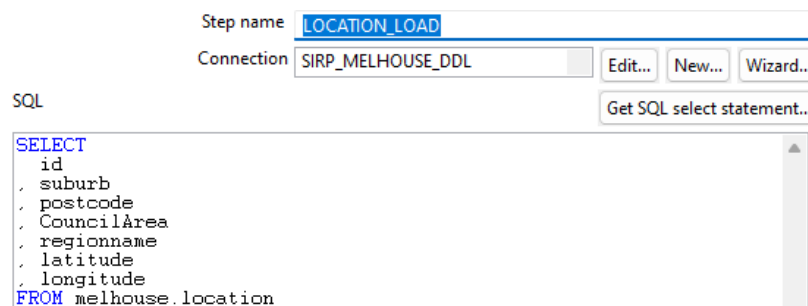
Slika 31: Deklaracija dimenzijske tablice 'Property'



Slika 32: ETL proces 'property' dimenzije

5.4. Dimenzijska tablica 'Location'

1. Dohvat podataka iz relacijske baze: Koristeći 'Table input' ponovno se učitavaju podaci iz relacijske baze podataka te ih se sortira i osigurava da su svi podaci točnog formata i vrste u 'Select Values'. Osiguravaju se tipovi podataka 'suburb' na String dužine 100, 'postcode' na Integer dužine 4, 'CouncilArea' na String dužine 100, 'regionname' na String dužine 100, 'latitude' na BigNumber dužine 12 i 'longitude' na BigNumber dužine 12. Svaki podatak kovertiramo iz binarnog formata u normalni kako bi izbjegli probleme sa procesom spajanja



Slika 33: Učitavanje podataka iz MySQL baze podataka

The screenshot shows a 'Select & Alter' window with a 'Meta-data' tab selected. It displays a table with 7 rows of fields to alter meta-data for. The table has columns: #, Fieldname, Rename to, Type, Length, Precision, and Binar.

#	Fieldname	Rename to	Type	Length	Precision	Binar
1	id		None	9	0	N
2	address		None	90		N
3	landsize		Integer	9	0	Y
4	buildingArea		String	50		Y
5	yearBuilt		String	50		Y
6	propertycount		Integer	9	0	Y
7	parking_area		String	50		Y

Slika 34: Formatiranje podataka iz MySQL-a

2. Dohvat podataka iz CSV datoteke: Putem 'CSV file input' se uzimaju podaci iz CSV datoteke. Sa prvim 'Select values' se obavlja selekcija podataka te se preimenuje i osigura vrsta i format podataka povučenih iz CSV-a. Sa 'Add sequence' elementom se generira atribut 'id' od zadnjeg poznatog id-a u

relacijskoj bazi te u drugom 'Select values' se osigurava točan redoslijed atributa za spajanje podataka. Redoslijed glasi: 'id' (Integer), 'suburb' (String), 'postcode' (Integer), 'CouncilArea' (String), 'regionname' (String), 'latitude' (BigNumber), 'longitude' (BigNumber).

Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to
1	Suburb	suburb	String	100		Y
2	Postcode	postcode	Integer	4		Y
3	CouncilArea		String	100		Y
4	Regionname	regionname	String	100		Y
5	Latitude	latitude	BigNumber	12	6	Y
6	Longitude	longitude	BigNumber	12	6	Y

Get fields to change

Slika 35: Formatiranje podataka iz CSV-a

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional)

Start at value

Increment by

Maximum value

Slika 36: Generiranje 'id' atributa za podatke iz CSV-a

Select & Alter Remove Meta-data

Fields:

#	Fieldname	Rename to	Length	Precision
1	id			
2	suburb			
3	postcode			
4	CouncilArea			
5	regionname			
6	latitude			
7	longitude			

Get fields to select

Edit Mapping

Slika 37: Sortiranje atributa iz CSV-a

- Merge i spremanje: Uz zadnjem koraku se spajaju dvije tablice po atributu 'id' i sortiraju se po 'Ascending' te se podatke prema postavljajući ključ 'id' i Fields se popunjava preostalim podacima – id, suburb, postcode, CouncilArea, regionname, latitude i longitude. Generira se tablica pomoću SQL funkcije te se podaci spremaju. U MySQL-u se briše *null* red.

Step name **Sorted merge 2 2 2**

Fields :

#	Fieldname	Ascending
1	id	Y

Slika 38: Merge na podacima iz SQL-a i CSV-a

Step name **LOCATION_DIM_CREATE**

Update the dimension? ☒

Connection **SIRP_MELHOUSE_DDL** Edit... New... Wizard...

Target schema **MELHOUSE_DDL** Browse...

Target table **LOCATION_DIM** Browse...

Commit size **100**

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all) **5000**

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	id	id

Technical key field **location_tk** New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field **version**

Stream Datefield

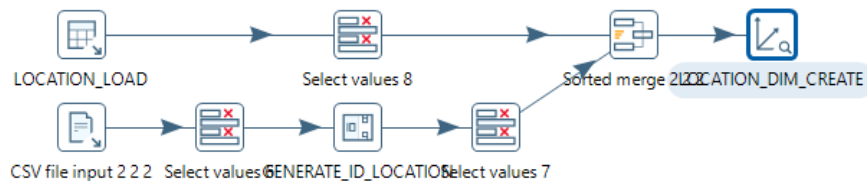
Date range start field **date_from** Min. year **1900**

Use an alternative start date? ☐ <Select Option>

Table date range end **date_to** Max. year **2199**

OK Cancel Get Fields SQL

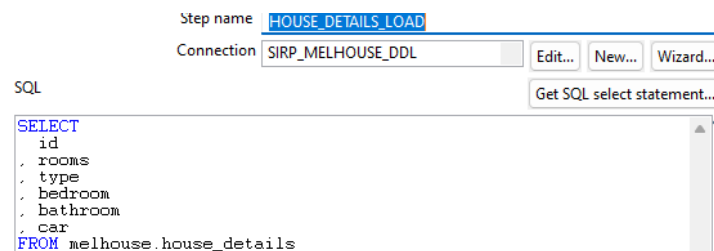
Slika 39: Deklaracija dimenzijske tablice 'Location'



Slika 40: ETL proces za 'location'

5.5. Dimenzijska tablica 'House_details'

1. Dohvat podataka iz relacijske baze: Koristeći 'Table input' ponovno se učitavaju podaci iz relacijske baze podataka te ih se sortira i osigurava da su svi podaci točnog formata i vrste u 'Select Values'. Osiguravaju se tipovi podataka 'rooms' na Number dužine 2, 'type' na String dužine 1 te 'bedroom', 'bathroom' i 'car' na Number dužine 3. Svaki podatak kovertiramo iz binarnog formata u normalni kako bi izbjegli probleme sa procesom spajanja



Slika 41: Učitavanje podataka iz MySQL baze podataka

2	car		Number	3	1	A
4	bedroom		Number	3	1	A
3	bathroom		Number	3	1	A
5	type		String	1		A
1	rooms		Number	5	0	A
6	Fieldname	Parameter	Type	Length	Precision	Binary to Normal

Slika 42: Formatiranje podataka iz MySQL baze podataka

2. Dohvat podataka iz CSV datoteke: Putem 'CSV file input' se uzimaju podaci iz CSV datoteke. Sa prvim 'Select values' se obavlja selekcija podataka te se preimenuje i osigura vrsta i format podataka povučenih iz CSV-a. Sa 'Add sequence' elementom se generira atribut 'id' od zadnjeg poznatog id-a u relacijskoj bazi te u drugom 'Select values' se osigurava točan redoslijed atributa za spajanje podataka. Redoslijed glasi: 'id' (Integer), 'rooms' (Number), 'type' (String), 'bedroom' (Number), 'bathroom' (Number), 'car' (Number).

Select & Alter

Remove

Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to No
1	Rooms	rooms	Number	2	0	Y
2	Type	type	String	1		Y
3	Bedroom	bedroom	Number	3	1	Y
4	Bathroom	bathroom	Number	3	1	Y
5	Car	car	Number	3	1	Y

Get fields to change

Slika 43: Formatiranje podataka iz CSV-a

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional)

Start at value

Increment by

Maximum value

[Help](#)

Slika 44: Generiranje atributa 'id'

#	Fieldname	Renname to	Length	Precision
0	cat			
2	pathroom			
4	bedroom			
3	type			
5	rooms			
1	id			

Edit Mapping

Get fields to select

Fields :

Select & Alter

Remove

Meta-data

Slika 45: Sortiranje atributa iz CSV-a

3. Merge i spremanje: Uz zadnjem koraku se spajaju dvije tablice po atributu 'id' i sortiraju se po 'Ascending' te se podatke sprema postavljajući ključ 'id' i Fields se popunjava preostalim podacima – rooms, type, bedroom, bathroom, car. Generira se tablica pomoću SQL funkcije te se podaci spremaju. U MySQL-u se briše *null* red.

Sorted merge

Step name Sorted merge 3

Fields :

#	Fieldname	Ascending
1	id	Y

Slika 46: Merge po atributu 'id'

Step name HOUSE_DETAILS_DIM_CREATE

Update the dimension? ☒

Connection SIRP_MELHOUSE_DDL Edit... New... Wizard...

Target schema MELHOUSE_DDL Browse...

Target table HOUSE_DETAILS_DIM Browse...

Commit size 100

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all) 5000

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	id	id

Technical key field house_details_tk New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field version

Stream Datefield

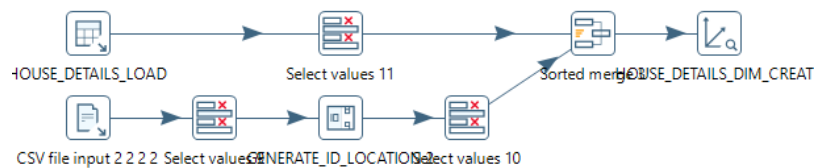
Date range start field date_from Min. year 1900

Use an alternative start date? ☐ <Select Option>

Table date range end date_to Max. year 2199

OK Cancel Get Fields SQL

Slika 47: Deklaracije dimenzijske tablice 'House_details'



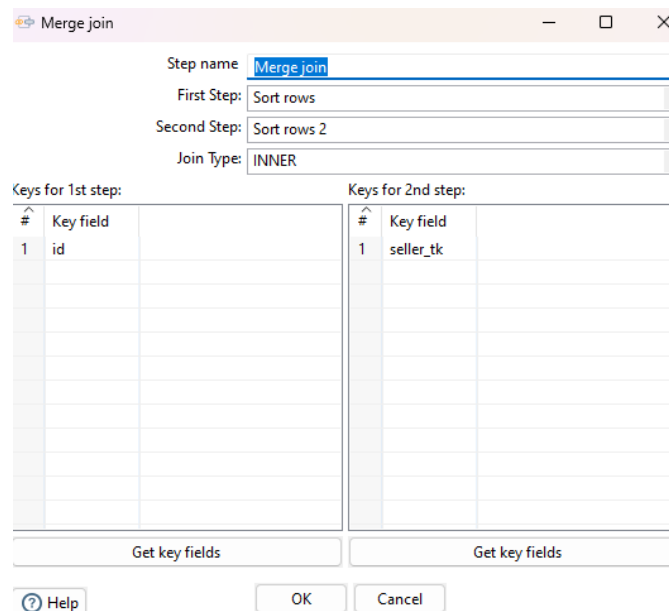
Slika 48: ETL proces za 'House_details'

5.6. Tablica činjenica 'Sale'

Za spajanje podataka iz MySQL baze podataka i CSV datoteke ponavlja se ETL proces korišten za dimenzijske tablice. Učitava se podatak 'price' koji se postavlja na tip podatka Number dužine 9.

U sljedećem koraku se koristi 'Merge Join' postavljen na 'INNER JOIN' kako bi se dimenzijske tablice spojile sa tablicom 'sale'. Kako bi se spojila dimenzijska tablica na tablicu činjenica koriste se atribut 'id' iz 'sale' tablice činjenica i tehnički ključ iz dimenzijske tablice.

Iz novodobivene tablice brišemo sve attribute osim 'id', 'seller_tk', 'sale_info_tk', 'property_tk', 'location_tk', 'house_details_tk' i 'price'. U zadnjem koraku koristimo ugrađenu funkciju SQL kako bi automatski stvorili tablicu činjenica po unesenim podacima u PDI alatu.



Slika 49: primjher joina tablice činjenica i 'seller'

Step name: Select values

Select & Alter Remove Meta-data

Fields to remove:

#	Fieldname
1	version
2	date_from
3	date_to
4	id_1
5	seller_name
6	version_1
7	date_from_1
8	date_to_1
9	id_2
10	dateOfSale
11	method
12	version_2
13	date_from_2
14	date_to_2
15	id_3
16	address
17	landsize
18	buildingArea
19	yearBuilt
20	propertycount
21	parking_area
22	version_3
23	date_from_3
24	date_to_3
25	id_4
26	suburb
27	postcode
28	CouncilArea
29	regionname
30	latitude
31	longitude
32	version_4
33	date_from_4
34	date_to_4
35	id_5
36	rooms
37	type
38	bedroom
39	bathroom
40	car

Get fields to remove

OK Cancel

Slika 50: Brisanje nepotrebnih podataka iz tablice činjenica

Step name: Table output

Connection: MELHOUSE_DIM_TEST Edit... New... Wizard...

Target schema: MELHOUSE_DDL Browse...

Target table: FACT_DIM Browse...

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options Database fields

Partition data over tables: ☐

Partitioning field:

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table:

Store the tablename field: ☒

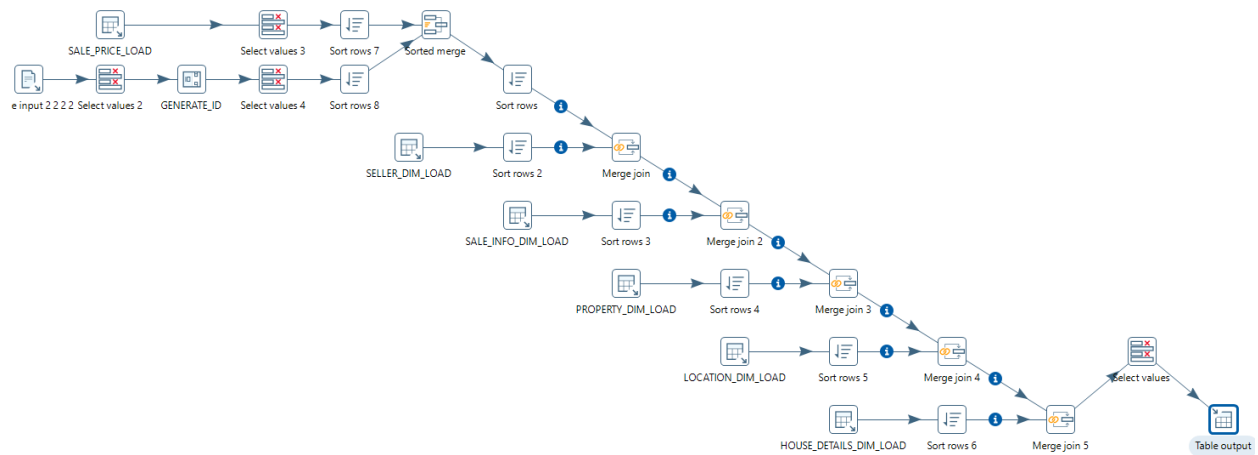
Return auto-generated key: ☐

Name of auto-generated key field:

Help OK Cancel SQL

Slika 51: Deklaracija tablice činjenica

5.6.1. Prikaz tablice činjenica



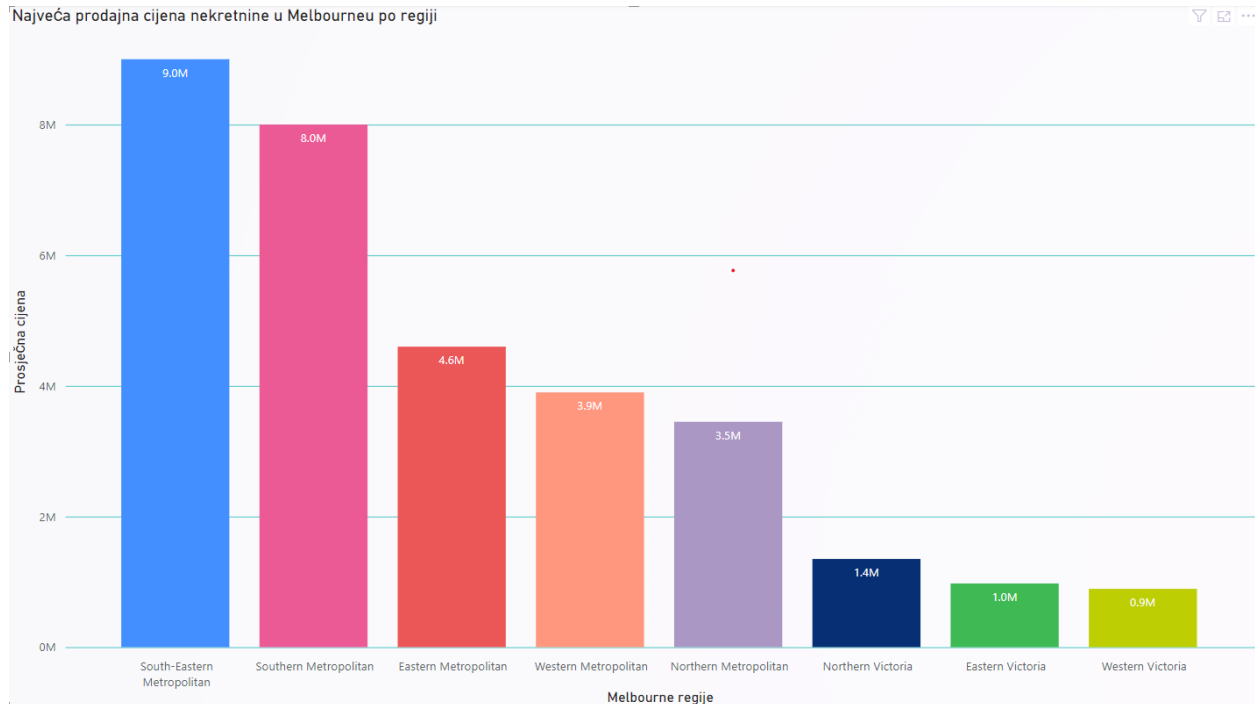
Slika 52: ETL proces tablice činjenica

6. Vizualizacija podataka

Za kraj nakon stvaranja dimenzijskog modela i izvođenja ETL procesa na podacima, vrijeme je predstaviti podatke u lako probavljiv oblik podataka, tj. vizualizirati ih u smislenim grafovima. Za vizualizaciju podataka koristimo Microsoft Power BI koji omogućava jednostavno stvaranje interaktivnih slide-a koristeći podatke pohranjene u dimenzijskom modelu. Svaki graf je spremljen u individualnom modelu.

Kao prvi korak se uspostavlja veza sa MySQL bazom podataka koja sadrži shemu dimenzijskog modela što je lagan proces pošto PBI podržava laganu vezu između alata i ostalih opširno korištenih tehnologija za skladištenje podataka.

6.1. Prikaz najveće prodaje po regiji

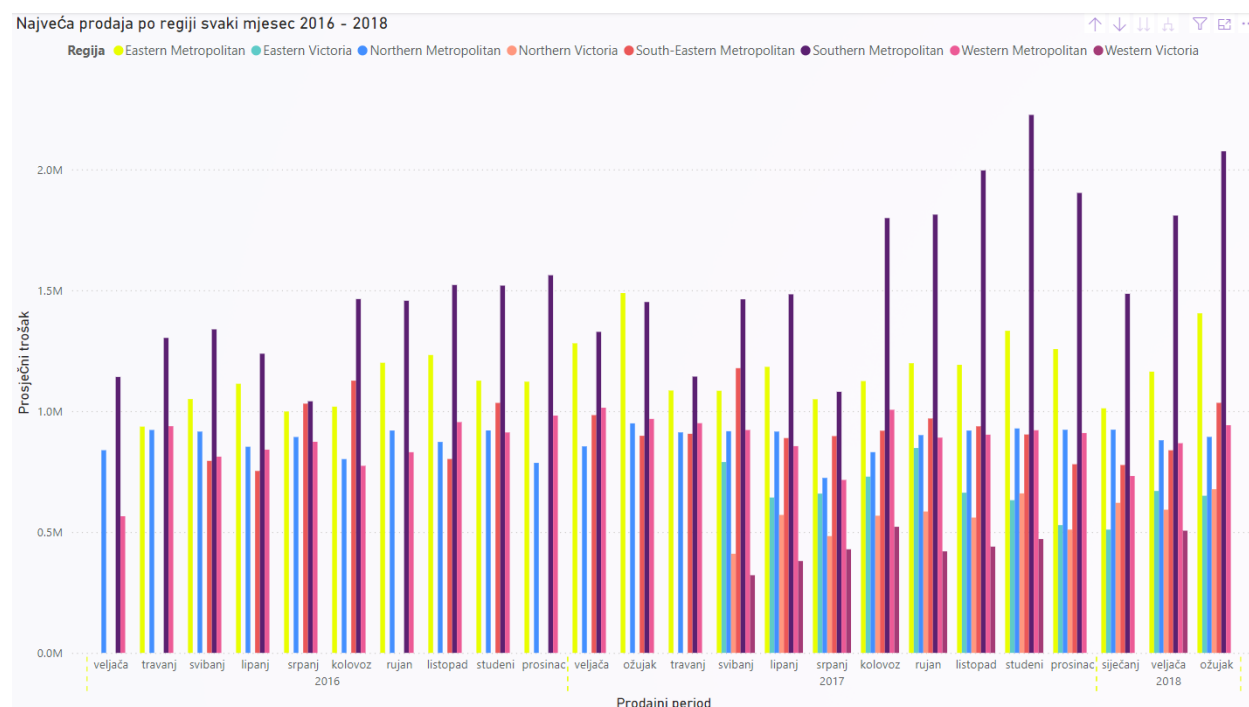


Slika 53: Najveća prodaja po regiji

Graf prikazuje najveću prodajnu cijenu nekretnine u svakoj određenoj regiji Melbourne-a. Ovaj graf se može koristiti kao indikator regija sa potencijalno najvrijednijim nekretninama te ciljati poboljšati aktivnost na tržištu baziranu baš na specifičnoj regiji. Također služi kao indikator potencijane vrijednosti i interesa kupaca za ostalim nekretninama u regiji

Sastoji se od grupiranog stupčastog grafikona gdje na x osi nalazimo 'regionname' a na y osi nalazimo $\max(\text{price})$ tj. najveću prodaju u atributu 'price' u tablici činjenica.

6.2. Prosječna cijena nekretnina po regiji u periodu od 2016. do 2018.

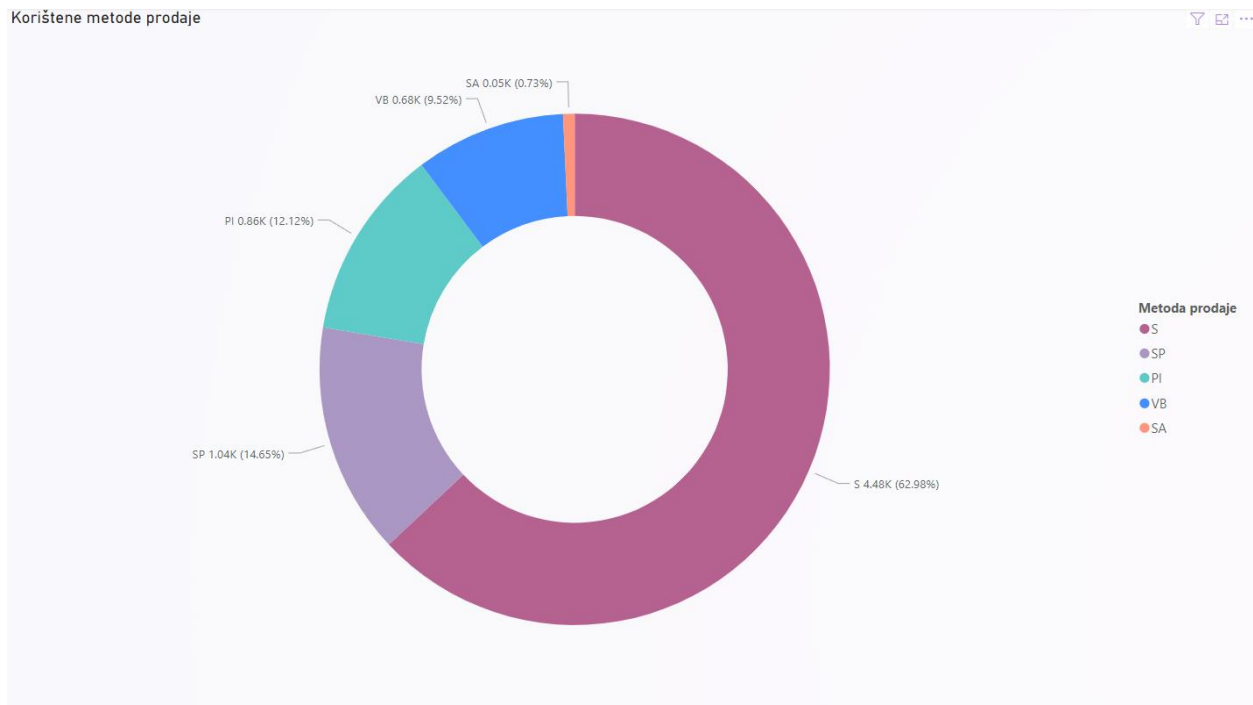


Slika 54: Graf prosječne cijene po regiji kroz vrijeme

Graf prikazuje prosječnu cijenu nekretnina raspodjeljene po regiji za svaki mjesec u periodu od veljače 2016. godine do ožujka 2018. godine te omogućava analizu sezonskih varijacija u prodajnim cijenama nekretnina. Prikazom podataka kroz mjesec u razdoblju od tri godine, može se uočiti kako se cijene mijenjaju tijekom različitih mjeseci te kako vremenski faktor utječe na tržište. Također se može uočiti trend u raznim regijama što daljnje podržava činjenice prikazane u prijašnjem grafu.

Sastoji se od grupiranog stupčastog grafikona gdje na x osi nalazimo 'dateOfSale' podijeljen na mjesec i godine. Na y osi nalazimo AVG(price) tj. prosječna cijena daljnje raspodijeljena po 'regionname'.

6.3. Prikaz korištenja metoda prodaja

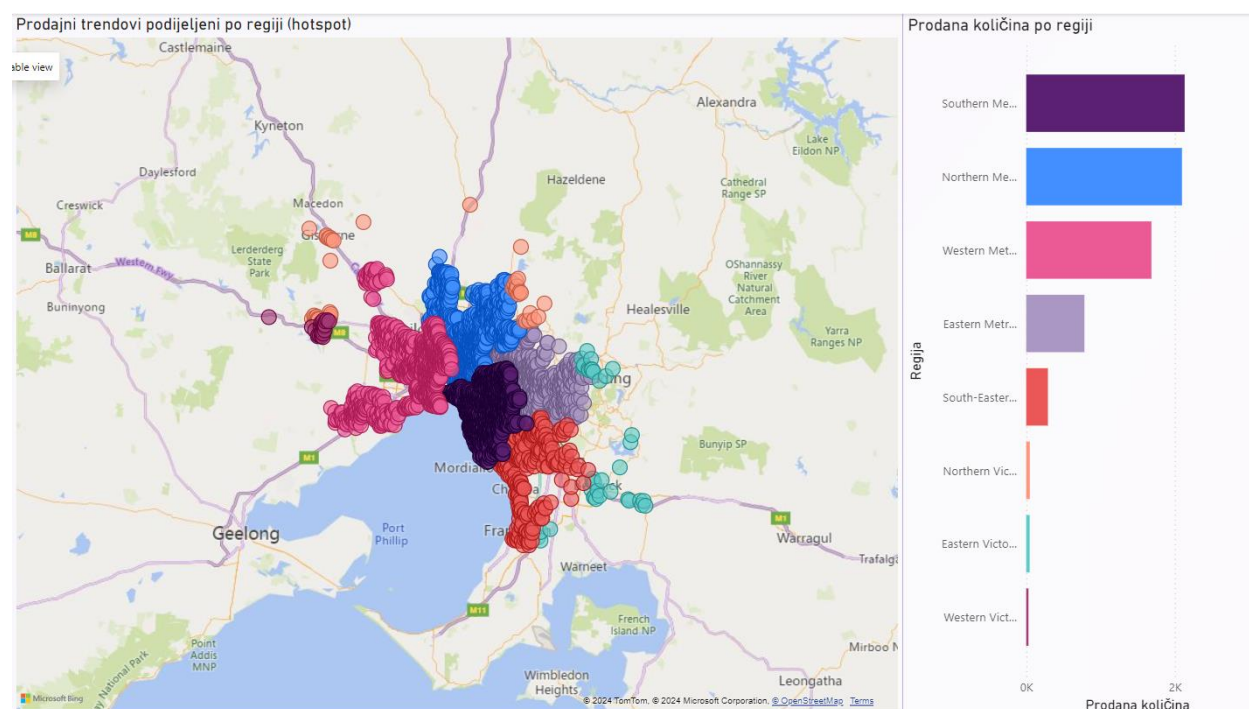


Slika 55: Graf količine korištenja metoda prodaje

Graf prikazuje postotak korištenja metode prodaje nekretnina u Melbourne-u. Uspoređujući udjele različitih metoda prodaja, moguće je uočiti dominantne metode te procijeniti učinkovitost istih. Ova informacija pomaže prilagođavanju strategije prodaje te je moguće procijeniti u kojem periodu procesa prodaje kupci “zagrizu”. Također se može dalje poboljšati za analizu utjecaja prodajnih metoda na vrstu nekretnine.

Sastoji se od prstenastog grafikona koji koristi `COUNT(fact_dim.id)` tj. broj prodaja vezanih za 'method' te računa postotak svake.

6.4. Prodajni trendovi podijeljeni po regiji

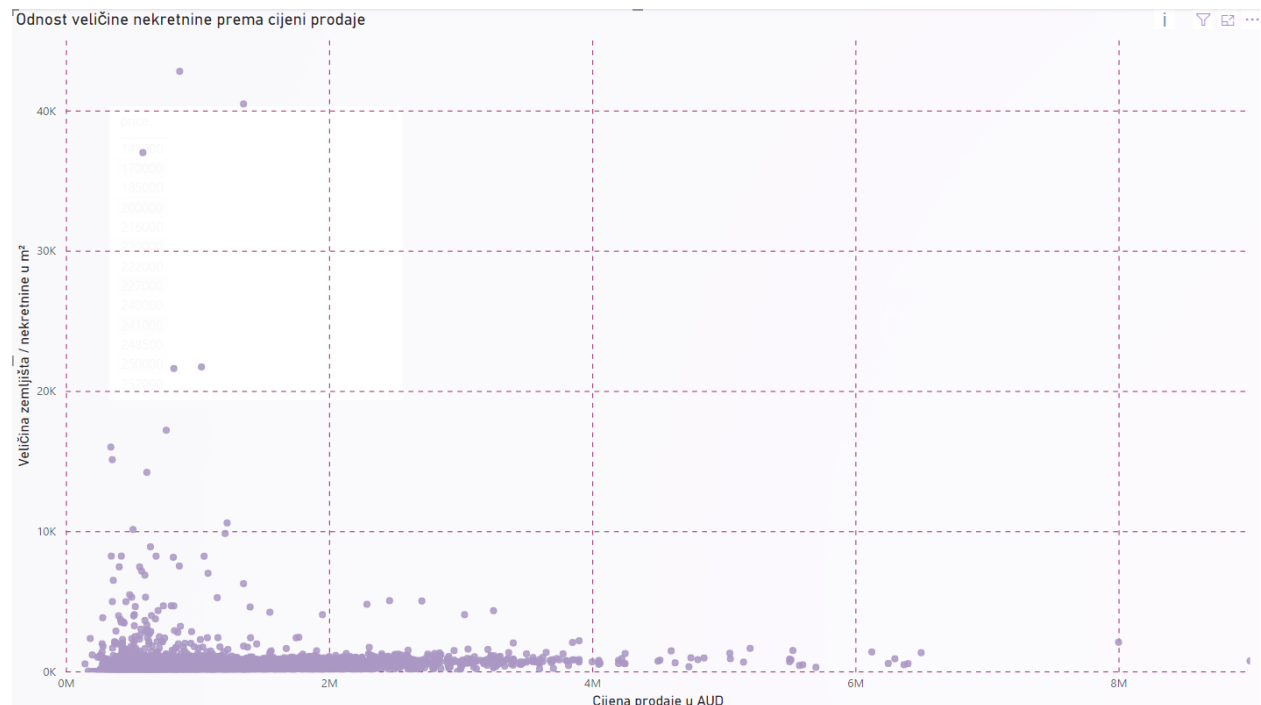


Slika 56: Prikaz prodajnih trendova na mapi raspodjeljene po regiji

Slide sadržava vizualni prikaz prodaja nekretnina po Melbourne-u. Koristi mapu koja služi kao heatmap koji pokazuje trendove tj. hotspotove u regiji gdje je interaktivno moguće uočiti ne samo trendove po regiji već i u samoj regiji zumirajući na mapi kako bi uočili da li neka specifična zona u regiji utječe na performansu. To dalje poboljšava prodajnu strategiju te pomaže u bolje informiranim poslovnim odlukama pokazujući na obećavajuće tržišne aktivnosti u određenoj zoni. Također na desnoj strani se nalazi količina prodanih nekretnina po regiji koja služi kao dodatni indikator trendova na tržištu.

Sastoji se interaktivne mape koja prima koordinate 'latitude' i 'longitude' gdje su rezultati podijeljeni po 'regionname' tj. regiji. Drugi graf se sastoji od grupiranog stupčastog grafikona koji sadrži COUNT(fact_dim.id) tj. količinu prodaja brojeći retke u tablici činjenica podijeljene dalje po regiji.

6.5. Odnos veličine zemljišta i cijene

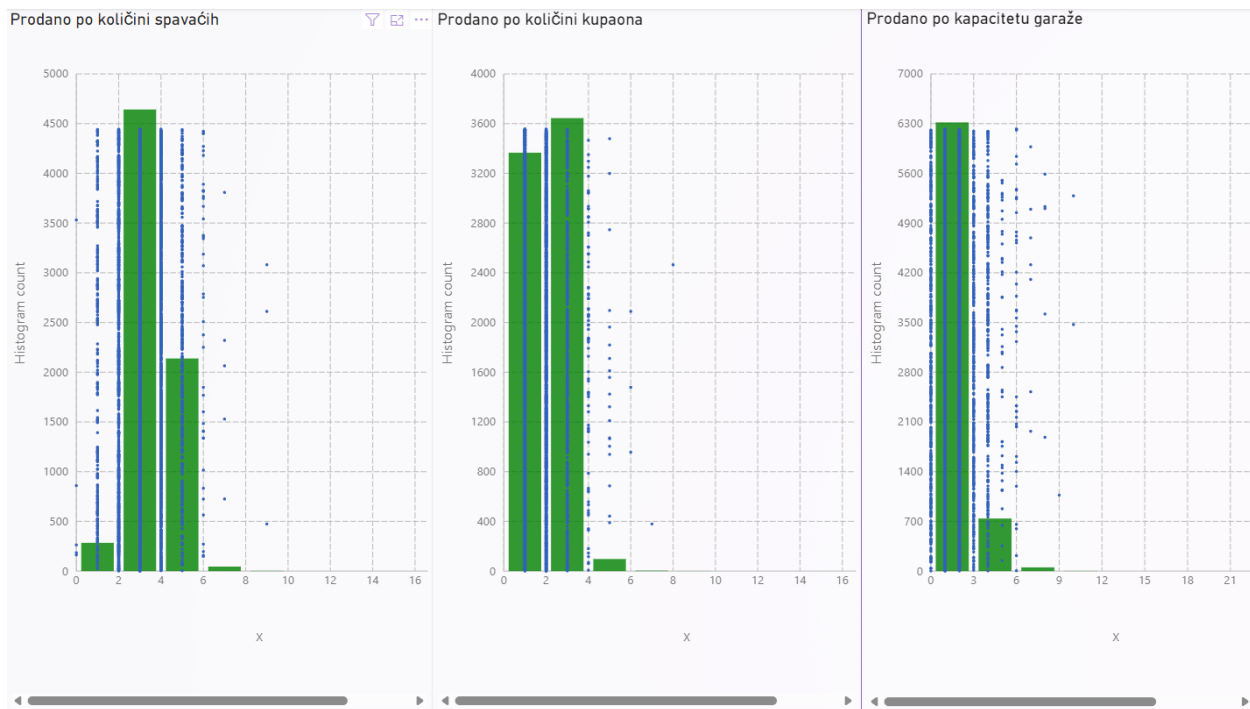


Slika 57: Prikaz odnosa cijene i veličine zemljišta

Graf prikazuje odnos između cijene i veličine zemljišta. Koristeći graf moguće je uočiti kako veličina zemljišta utječe na cijenu nekretnine, no iako se može koristiti kao indikator vrijednosti neke nekretnine ne uračunava ostale faktore poput vrijednosti građevine koja se nalazi na nekretnini te razne komoditete koji dolaze sa ponudom. Također je važno uočiti da velike mjere ne prate neki specifični trend već kod velikih prodaja varijacija komoditeta je velika a količina podataka je nista te napraviti preciznu tj. dobro informiranu odluku po gore vidljivim podacima je teško.

Sastoji je od raspršenog grafikona koji prima 'price' tj. cijenu na x osi i 'landsize' tj. veličinu zemljišta na y osi.

6.6. Praćenje trenda prodaje po dostupnim komoditetima

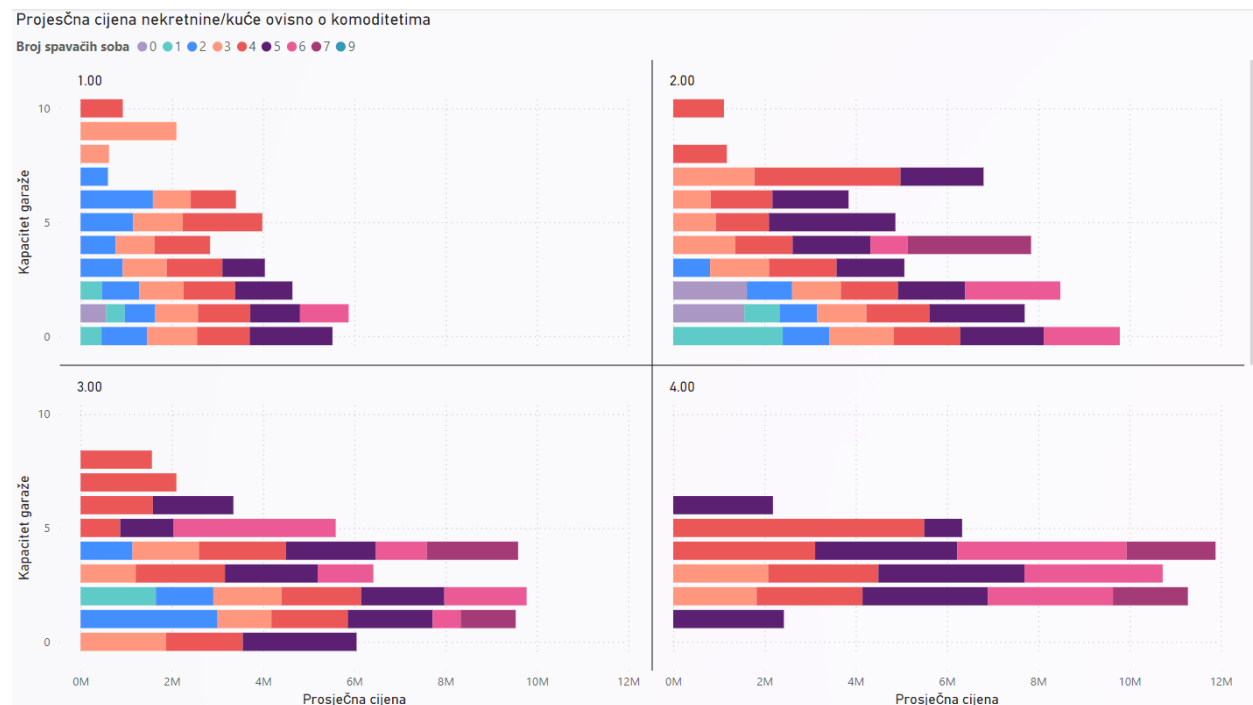


Slika 58: Prikaz utjecaja komoditeta na prodajnu količinu

Grafovi prate prodajne trendove ovisno o dostupnim komoditetima u 3 različina grafa. Kupci često gledaju dostupne komoditete sa nekretninom zbog raznih faktora poput veličine obitelji, željena količina prostora... Ovim grafovima je moguće analizirati utjecaj određenog praćenog komoditeta na odluku kupca te se dostupnim informacijama može fokusirati na specifični dio demografije Melbourne-a te dovesti do boljih odluka o ulaganju.

Sastoji se od 3 različita histograma koji na x osi sadržavaju 'bedroom', 'bathroom' i 'car', tj. broj spavaćih, kupaoonica i kapacitet garaže a na y osi se nalazi COUNT(fact_dim.id) tj. količina prodaja u tablici činjenica.

6.7. Cijena nekretnine ovisno o dostupnim komoditetima

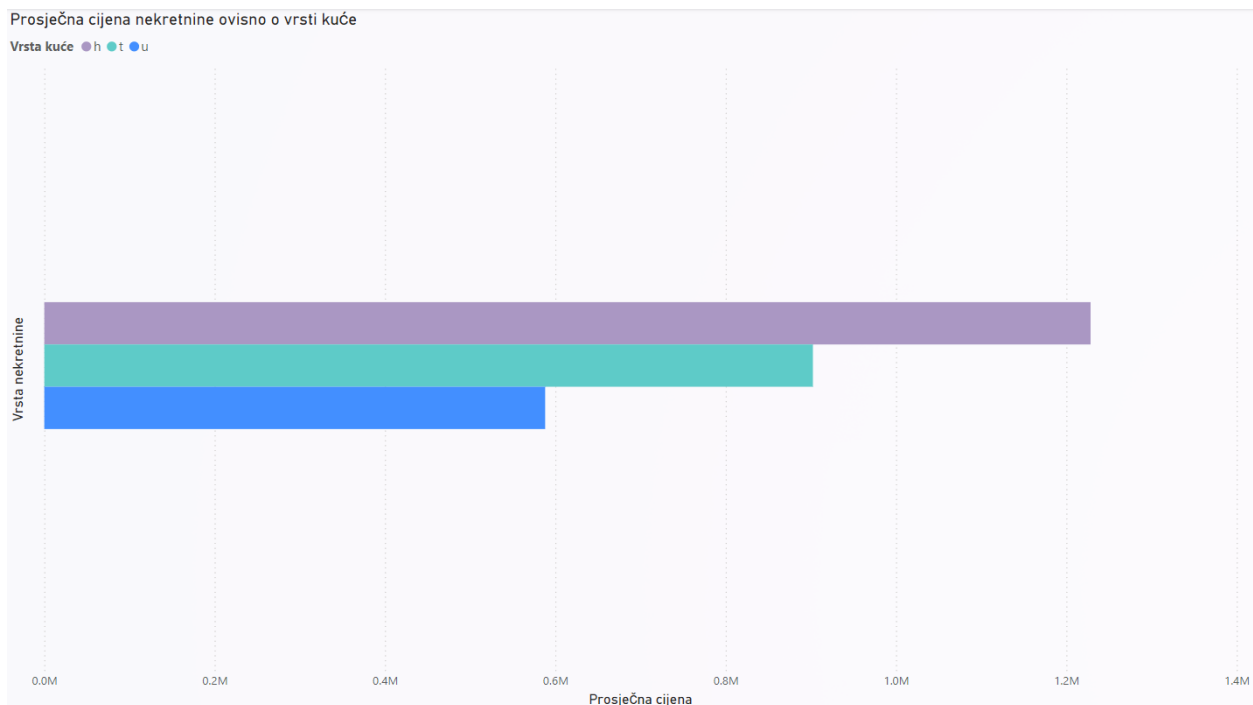


Slika 59: Prikaz utjecaja komoditeta na prodajnu cijenu

Graf prikazuje utjecaj raznih količina komoditeta na cijenu prodaje nekretnine. Ovaj graf nam nadograđuje pogled nad utjecajem komoditeta na popularnost nekretnina te nam omogućava pratiti trendove i postavljati kompetitivne cijene za nekretninu na tržištu.

Sastoji se od naslaganog stupčastog grafikona koji AVG(price) tj. prosječnu cijenu određuje po dostupnim komoditetima sa nekretninom. Iako nam graf daje puno informacija, mali setovi podataka mogu negativno utjecati na rezultate u velikim prodajama koje su puno rjeđe u manjim nekretninama.

6.8. Prosječna cijena ovisno o vrsti nekretnine

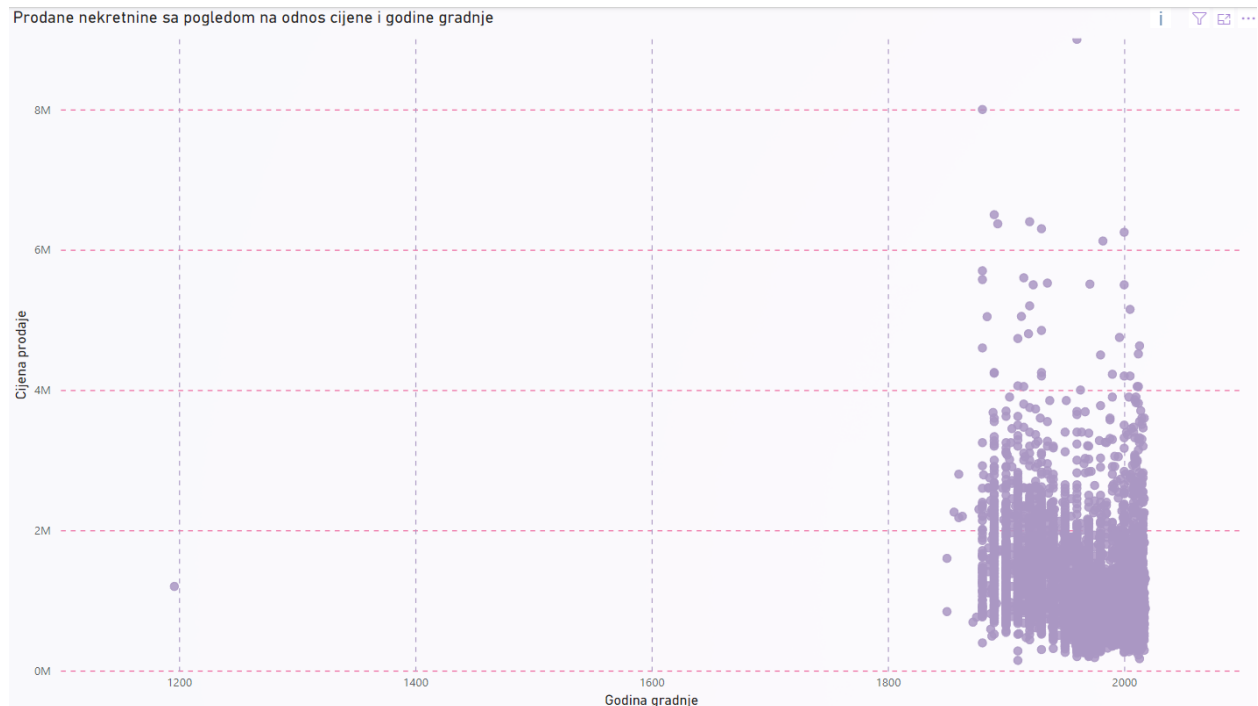


Slika 60: Prikaz prosječne cijene nekretnine ovisno o vrsti

Graf nam prikazuje trendove i popularnost određenih vrsta nekretnina gledajući prosječnu cijenu na tržištu. S time da su individualne i gradske kuće često veće od stanova rezultat je donekle i očekivan iako to ovisi od zone do zone gdje u velikim metropolama dobro lociran stan u centru može lagano premašiti vrijednost i najlijepših vila.

Sastoji se od grupiranog stupčastog grafikona koji na x osi prati AVG(price) tj. prosječnu cijenu podijeljenu po 'type' tj. vrsti nekretnine.

6.9. Utjecaj godine gradnje na cijenu

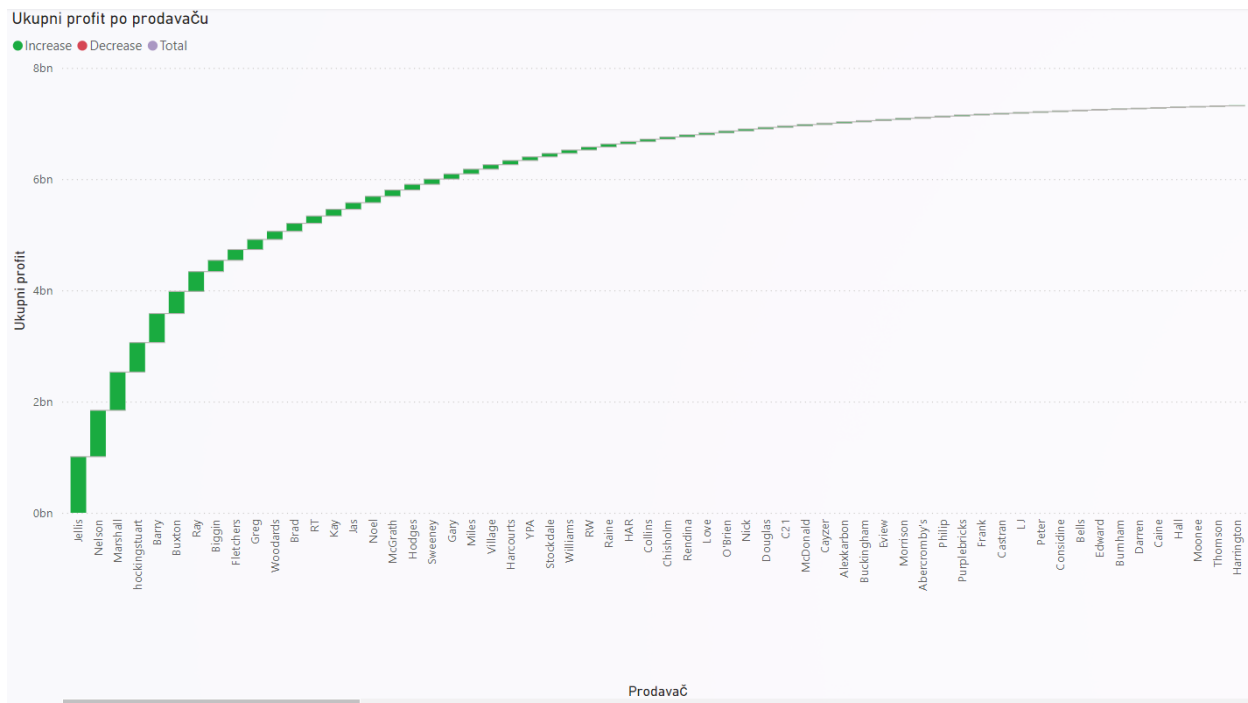


Slika 61: Prikaz odnosa godine gradnje i cijene

Graf prikazuje odnos cijene nekretnine i godine gradnje nekretnine. S time da starije kuće zahtjevaju renovacije te su često napravljene u starijim i često lošijim tehnikama građevine, kupci često nisu zainteresirani zbog velike količine novca koje se često ulaže kako bi se osvježila i nadogradila građevina. Starije nerenoviranje građevine također mogu dovesti do legalnih problema u većim gradovima koji se pridržavaju smjernica izgleda i očuvanja okoliša.

Sastoji se od raspršenog grafikona koji na x osi predstavlja 'yearBuilt', tj. godinu gradnje i na y osi 'price', tj. cijenu prodaje nekretnine.

6.10. Performansa prodavača

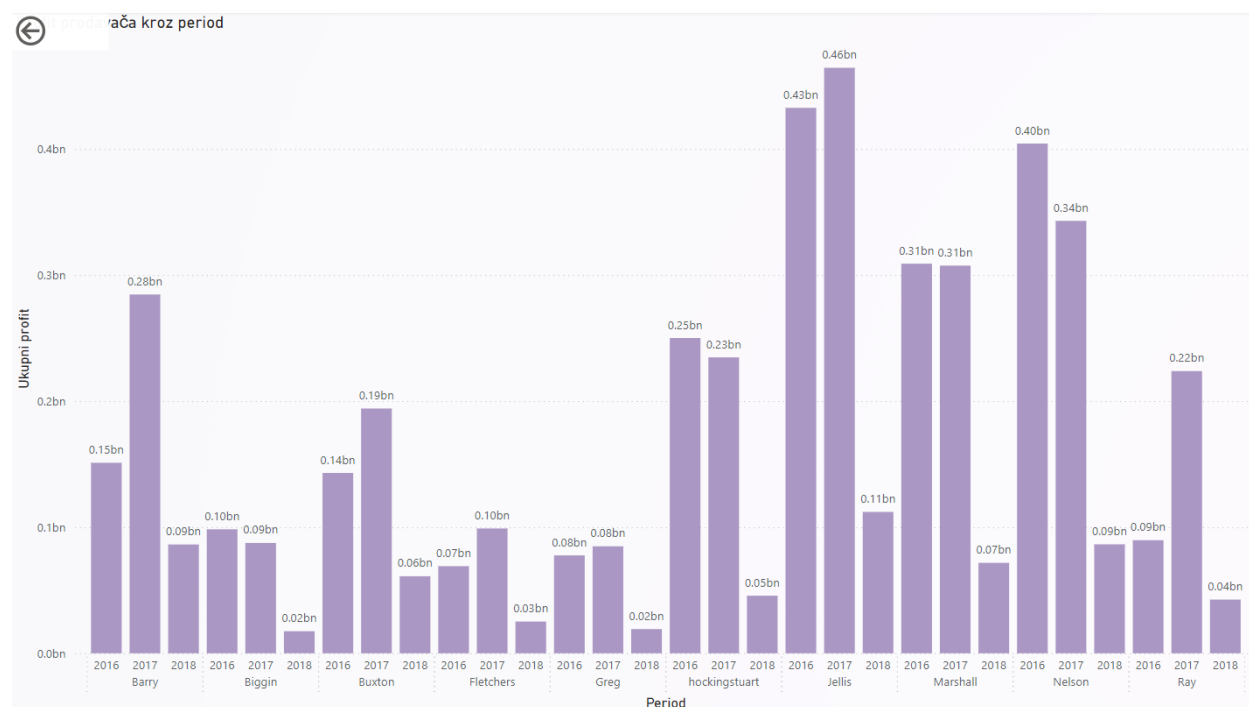


Slika 62: Prikaz ukupnog doprinosa prodavača

Graf prikazuje ukupni profit svakog prodavača u datasetu. Ukupni profit je dobar indikator performanse prodajnog agenta te prateći njihove trendove možemo uočiti korištene strategije i poslovne odluke koje su kontribuirale uspjehu ili neuspjehu prodajnog agenta. Na kraju grafa nam je dostupna i ukupna količina novca potrošena na tržištu što nam dalje daje bolji uvid o kvaliteti implementiranih strategija. Cijeli graf je vidljiv na interaktivnom sučelju u Power BI alatu.

Sastoji se od slap grafikona koji za kategoriju koristi 'seller_name', tj. ime prodavača i na y osi koristi SUM(price), tj. ukupnu količinu novca te se prikazuje po ukupnoj količini novca dobivenoj ili izgubljenoj po prodajnom agentu.

6.11. Performansa prodavača kroz vrijeme



Slika 63: Prikaz performanse najboljih prodavača

Graf nam prikazuje performansu deset najboljih prodavača u definiranom periodu. Omogućuje dublji pogled u prijašnji graf i informacije koje pruža. Sa dodatkom vremenskog perioda možemo također usporediti stanje tržišta u vremenu i osigurati se da uspješni val prodaja nije bio atributiran anomaliji u tržištu. Vremenski period se može dalje razvući sa opcijom implementacije mjeseca ili dana za precizniji pogled kroz vrijeme.

Sastoji se od grupiranog stupčastog grafikona koji na x osi prati 'seller_name' i 'dateOfSale' koji predstavljaju prodavača kroz vrijeme i SUM(price) tj. sumu cijene koja predstavlja ukupni profit prodavača.

7. Zaključak

Pomoću napisanog seminara možemo uočiti da sa točnim alatima i modernim implementacijama u poslovanju možemo dovoditi kompetitivne i precizne poslovne odluke te dalje usavršiti poslovnu strategiju. Cilj je dokazati da informacija prenesena u znanje dovodi do puno manje grešaka te u svijetu gdje se informacije kreću zastrašujućom brzinom, imati pravu informaciju u pravo vrijeme smanjite rizik istih. Sam proces modeliranja podataka te pretvorba u informaciju je prilično zahtjevan. Znanje i iskustvo korištenja raznih alata poput PDI, MySQL-a i prograskog jezika poput Pythona koji se sve više i više koristi kao osnova za obradu podataka je sve vrijednije iz dana u dan te postaje obavezno kako bi ostali kompetitivni u poslovanju.

8. Literatura

Kimball, R., Ross, M. (2013). The data warehouse toolkit: The definitive guide to dimensional modeling. John Wiley and Sons.

Sharda, R., Delen, D., Turban, E. (2016). Business intelligence, analytics, and data science: a managerial perspective. Pearson.

Oreški, G., Prezentacije i ostali materijali na e-učenju.