**Dokumentatsioon ja kokkuvõte.**

**ERD – Entity Relational Diagram:A document with text on it

Description automatically generated**

**Kokkuvõte tehtud töödest ja õppetundidest:**

Selle projekti käigus õppisin palju relatsiooniliste andmebaaside loomise ja haldamise kohta, sealhulgas andmete struktureerimise, päringute koostamise ja andmebaasimudelite vaheliste seoste loomise kohta. Projekti eesmärgiks oli luua kinnisvara andmebaas, mis võimaldaks hallata kinnisvaraobjekte, turuosalisi ja tehinguid. Eesmärk oli luua süsteem, mis võimaldaks hõlpsasti lisada uusi kirjeid, muuta olemasolevaid ja analüüsida kinnisvaraturu andmeid.

Alguses plaanisin kasutada ja importida reaalseid andmeid Maa-ameti, Statistikaameti ja Notarite Koja kinnisvarastatistika andmebaasidest. Kahjuks selgus, et nende allikate andmeid ei olnud võimalik otse importida või need ei sisaldanud piisavalt üksikasjalikke kirjeid, et neid projekti kontekstis kasutada. Seetõttu otsustasin luua dummy dataset-i, mis jäljendas realistlikke kinnisvaratehinguid ja nende atribuute, et viia läbi erinevaid päringuid.

**Õppetunnid Andmebaaside Kohta:**

1. **Andmebaasi Disain ja Seosed**:
   * Õppisin, kuidas modelleerida andmestruktuure, mis esindavad reaalse maailma objekte ja nende omavahelisi seoseid. Näiteks tuli läbi mõelda, kuidas luua kinnisvara, linnade ja maakondade vahelised seosed. See oli oluline, et vältida andmete dubleerimist ja tagada nende terviklikkus.
   * Relatsiooniliste andmebaaside põhimõtted, nagu välisvõtmed ja tabelite vahelised seosed, aitasid mul luua loogilise andmebaasi struktuuri.
2. **Keerukad Päringud ja Andmeanalüüs**:
   * Projekti raames koostasin mitmesuguseid keerulisi päringuid, et kuvada näiteks tehinguid teatud hinnavahemikus, kinnisvaraobjekte konkreetsetes piirkondades või analüüsida tehingute arvu aastate lõikes.
   * Õppisin, kuidas koostada koondpäringuid ja kasutada erinevaid SQL-i funktsioone, nagu COUNT, AVG, MAX ja MIN, et saada ülevaade andmetest.
3. **Andmete Importimise ja Puhastamise Väljakutsed**:
   * Kuigi algselt oli plaanis importida andmed ametlikest allikatest, andis see mulle kogemuse, kuidas reaalse maailma andmed ei pruugi olla otse kasutatavad. Näiteks ei olnud andmed piisavalt detailsed või nende struktuur ei sobinud minu loodud mudelitega.
   * Selle probleemi lahendamine dummy dataset-i loomise kaudu võimaldas mul keskenduda andmebaasi ülesehituse ja päringute testimisele.

**Miks Valisin PostgreSQL-i:**

Valisin PostgreSQL-i, sest varasemalt olen veidi töötanud SQLite ja MySQL-ga ning tahtsin nüüd proovida PostgreSQL-i, et omandada kogemusi uue andmebaasisüsteemiga. Lisaks kasutasime PostgreSQL-i ka tundides, mis andis mulle kindluse sellega jätkata ja süvendada õpitut.

**PostgreSQL-il on ka mõningaid eeliseid teiste alternatiivide ees:**

* Komplekssete päringute tugi: PostgreSQL toetab keerulisi päringuid ja andmetüüpide kasutamist, mis teeb selle sobivaks projektide jaoks, kus on vaja teha süvitsi andmeanalüüsi või keerulisi seoseid.
* Avatud Lähtekood ja Lai Toetus: PostgreSQL on avatud lähtekoodiga ja sellel on suur kasutajaskond ning kogukond, mis teeb lihtsamaks abi ja juhendite leidmise, kui tekib probleeme.

Need põhjused tegid PostgreSQL-i valiku loogiliseks ning pakkusid mulle võimalust laiendada oma andmebaasidega seotud teadmisi ja oskusi.

**Projekti Ülesehitus:**

1. **Andmebaasimudelid ja Nende Seosed**:
   * **Maakond ja Linn**: Maakond on ülemine tasand ja Linn kuulub vastavasse maakonda (1 to Many seos). See võimaldas luua hierarhilise struktuuri, kus iga linn viitab oma maakonnale.
   * **Kinnisvara ja Linn**: Iga kinnisvaraobjekt kuulub kindlasse linna, mis on seotud maakonnaga (1 to many). See aitab geograafiliselt korraldada kinnisvaraobjekte ja nende andmeid.
   * **Klient, Maakler ja Tehing**: Klientidel ja maakleritel on tehingutes olulised rollid. Klient võib olla nii ostja kui ka müüja (Many to Many seos). Maaklerid on seotud tehingutega, olles ühe kinnisvaratehingu maakleriks (1 to Many seos).
   * **Tehing ja Kinnisvara**: Iga tehing on seotud konkreetse kinnisvaraobjektiga. See seos on oluline, et jälgida, millised kinnisvaraobjektid on müüdud või üüritud. Igat kinnisvara võib aastate jooksul mitu korda müüa seega iga kinnisvara kohta võib olla mitu tehingut (1 to Many seos).

Many-to-Many seoste puhul kasutasin vahetabeleid, mille django loob automaatselt. (Tehing\_Ostja ja Tehing\_Müüja).

1. **Päringute ja Analüüsi Rakendamine**:
   * Kasutasin keerulisi koondpäringuid, et kuvada tehingute arvu aastate lõikes, keskmisi ruutmeetrihindu ja kinnisvarahindu erinevate kinnisvaratüüpide lõikes.
   * Kasutasin Django ORM-i, et hallata päringuid ja vähendada vajadust otsese SQL-i kirjutamiseks, mis muutis koodi loetavamaks ja hooldatavamaks.
2. **Kasutatud Tehnoloogiad**:
   * **Django ja PostgreSQL**: Nende kahe tööriista kombinatsioon võimaldas luua veebipõhise rakenduse koos relatsioonilise andmebaasiga.
   * **Python**: Andmetöötluseks ja mudelite defineerimiseks, samuti päringute ja andmete kuvamise loogika rakendamiseks.
   * **pgAdmin**: Andmebaasi haldamiseks ja testimiseks.
   * **ChatGPT**: Abi saamiseks keerulisemate andmebaasi- ja koodiprobleemide lahendamisel ning juhiste ja nõuannete saamiseks.
   * **Javascript, HTML ja CSS**: Päringute visualiseerimiseks ning veebilehitsejas funktsionaalsuse loomiseks.
   * **Github**: Repositoorium lähtekoodi ja failide hoidmiseks ja jagamiseks.

**Kokkuvõte:**

Projekti käigus saavutasin tugeva arusaama relatsiooniliste andmebaaside ja nende haldamise kohta, samuti nende integreerimisest Django veebirakendusse. Andmebaasi disain ja päringute koostamine olid projekti põhifookuses, pakkudes väärtuslikke õppetunde, mida saan kasutada ka tulevikus. Algsete andmete kogumise ja importimise raskuste tõttu pidin kohanema ja looma dummy dataset-i, mis pakkus siiski võimaluse praktiliselt andmebaasidega töötada. PostgreSQL valimine osutus õigeks otsuseks, kuna see pakkus vajalikku paindlikkust ja funktsionaalsust ning aitas mul paremini mõista relatsiooniliste andmebaaside toimimist.

Kõik koodid, ERD pilt, lähteülesanne ja andmebaasi tagavara ja requirements.txt failid on lisatud järgmisesse Github-i reposse. Dummy dataset loomise lähtekoodi lisasin Django realestate projekti kausta (samas, kus on ka manage.py).

<https://github.com/RSylla/Django_realestate_app>