

ԱՎՏՈՄԵՔԵՆԱՆԵՐԻ ՇՈՒԿԱՅԻ ՏՎՅԱԼՆԵՐԻ ԲԱզայի ՆԱԽԱԳԾՈՒՄ և ՎԵՐԼՈՒԾՈՒԹՅՈՒՆ

1. ՆԵՐԱԾՈՒԹՅՈՒՆ

Այս նախագիծը իրականացվել է որպես «Տվյալների բազաների կառավարում» (DBMS) առարկայի ավարտական աշխատանք:

Նպատակն է իրական տվյալների հիման վրա իրականացնել տվյալների բազայի նախագծման ամբողջական գործընթաց՝ սկսած տվյալների հավաքումից մինչև վերլուծական SQL հարցումների իրականացում:

Նախագիծը ներկայացնում է ավտոմեքենաների շուկայի տվյալների բազա՝ հիմնված List.am կայքի «Ավտոմեքենաներ» բաժնից հավաքված տվյալների վրա:

2. ՏՎՅԱԼՆԵՐԻ ՀԱՎԱՔՈՒՄ (Data Collection)

Տվյալների հավաքման փուլը իրականացվել է Python լեզվով՝ օգտագործելով սքրեյփինգի մեթոդ:

Սքրեյփինգի կողմանը տեղակայված է նախագծի `list_car_scraper-main/` պանակում:

Այս պանակում գտնվող սքրիպտը (`list_car_scraper.py`) ավտոմատ կերպով հավաքում է ավտոմեքենաների հայտարարությունների տվյալները List.am կայքից:

Սքրեյփինգի ընթացքում միաժամանակ իրականացվում է նաև տվյալների նախնական մշակում՝

- բացակայող արժեքների կարգավորում,
- տվյալների տիպերի վերափոխում,
- ոչ վավեր կամ դատարկ արժեքների մաքրում:

Սքրեյփինգի արդյունքը պահպանվում է մեկ ամբողջական CSV ֆայլում՝ `car_data.csv`, որը արդեն պատրաստ է հետագա վերլուծության և տվյալների բազայի մեջ ներմուծման համար:

3. ՏՎՅԱԼՆԵՐԻ ՄՇԱԿՈՒՄ և ԲԱԺՎԱՆՈՒՄ

Սքրիպտած ամբողջական `car_data.csv` ֆայլը հետագայում մշակվել է Python-ի միջոցով և բաժանվել է մի քանի տրամաբանական առյուսակների՝ տվյալների բազայի նորմալիզացիայի նպատակով:

Այս քայլը իրականացվել է `scripts/` պանակում գտնվող սքրիպտներով:

Ստացված CSV ֆայլերն են.

- `core.csv` – պարունակում է հիմնական listing-ի ինֆորմացիան
- `vehicle.csv` – ավտոմեքենայի մակարդակը և մոդել
- `pricing.csv` – գին, տարեթիվ, վազք
- `specs.csv` – շարժիչի և տեխնիկական բնութագրեր
- `appearance.csv` – գույն, ինտերիեր, արտաքին հատկանիշներ
- `status.csv` – մաքսազերծման և վիճակի տվյալներ
- `feature.csv` – ավտոմեքենաների ինսրավոր լրացուցիչ հատկությունների բառարան
- `listing_feature.csv` – միջանկյալ առյուսակ many-to-many կապի համար

Այս բաժանումը ապահովում է տվյալների բազայի նորմալիզացիան առնվազն մինչև **3-րդ նորմալ ձև (3NF)** և կանխում է տվյալների կրկնությունը:

4. Տվյալների բազայի նախագծում (Database Design)

Տվյալների բազան ստեղծվել է PostgreSQL միջավայրում:

Բոլոր SQL հրամանները գտնվում են `sql/schema.sql` ֆայլում:

Նախագծման հիմնական առանձնահատկությունները.

- Ստեղծված է առանձին schema՝ `car_market`, ոչ թե public
- Բոլոր առյուսակները ունեն առաջնային բանալի (Primary Key)
- Օգտագործված են արտաքին բանալիներ (Foreign Key) առյուսակների միջև կապերի համար
- `core` առյուսակը հանդիսանում է կենտրոնական entity
- Մնացած առյուսակները կապված են `core`-ի հետ 1:1 հարաբերությամբ
- `feature` և `core` առյուսակների միջև իրականացված է many-to-many կապ՝ `listing_feature` միջանկյալ առյուսակի միջոցով
- Կիրառված է partitioning ըստ `year` դաշտի՝ վերլուծական հարցումների արդյունավետության բարձրացման համար
- Ստեղծված են համապատասխան ինդեքսներ query-ների արագ կատարման նպատակով

- Կիրառված են CHECK, NOT NULL, UNIQUE և DEFAULT սահմանափակումներ

Տվյալների բազայի կառուցվածքը ներկայացված է ER Diagram-ում ([docs/ER_Diagram.png](#)):

5. Տվյալների ներմուծում (Data Insertion)

Տվյալների ներմուծումը իրականացվել է Python լեզվով՝ օգտագործելով [psycopg2](#) գրադարանը:

Ներմուծման սքրիպտը գտնվում է [scripts/data_insertion.py](#) ֆայլում:

Ներմուծման գործընթացը ներառում է.

- CSV ֆայլի ընթերցում
- Տվյալների վերջնական մաքրում և վավերացում
- Batch insert մեխանիզմ տվյալների արագ ներմուծման համար
- Transaction management՝ COMMIT և ROLLBACK
- Սխալների գրանցում (logging)
- Տվյալների բազայի մուտքային տվյալների օգտագործում environment variables-ի միջոցով (ոչ hardcoded)

Այս մուտքումը ապահովում է տվյալների ամբողջականությունը և անվտանգությունը:

6. SQL հարցումներ և վերլուծություն (Queries & Analysis)

SQL հարցումները գտնվում են [sql/queries.sql](#) ֆայլում:

Այս ֆայլը պարունակում է վերլուծական և բիզնես տրամաբանությամբ հարցումներ, որոնք ցույց են տալիս PostgreSQL-ի առաջարկել հնարավորությունների կիրառումը:

Օգտագործված են հետևյալ տեխնիկաները.

- JOIN-ներ (INNER, LEFT)
- Aggregation (COUNT, AVG, MIN, MAX)
- GROUP BY և HAVING
- BETWEEN, IN, ILIKE
- Window functions (ROW_NUMBER, LAG)
- CTE-ներ (WITH)
- CASE WHEN պայմանական տրամաբանություն

- COALESCE և NULLIF
- Partition-aware հարցումներ
- EXPLAIN ANALYZE՝ query-ների կատարման պլանի վերլուծության համար

Յուրաքանչյուր հարցում ունի մեկսաբանություն և հստակ բիզնես նպատակ, օրինակ՝ միջին գների համեմատություն ըստ տարիների, ավտոմեքենաների տարածված մակարդակի վերլուծություն, գների դինամիկա և այլն:

DBMS Final Project – 2026