Proiect baze de date

-Service auto-

1) Posesorii de automobile descriu experienta pe care o au la service-urile auto din mai multe zone, in review-uri. Mai multi clienti ai acelui service descriu experienta si dau o nota intre 1 si 10 service-ului si mecanicului care a lucrat la masina, specificand marca, modelul si anul autovehiculului sau. Un mecanic poate acumula mai multe note de-a lungul vietii, in functie de service-ul unde a lucrat. Un mecanic lucreaza la mai multe masini in cursul carierei, asa cum o masina este reparata de mai multi mecanici. Nota fiecarui service este anuntata anual, pentru fiecare judet, pe site-ul Top Service. Tot in site se vor gasii detalii despre preturi, pe o scara de la 1 la 4, pentru fiecare marca la care service-ul respectiv lucreaza.

Site-ul Top Service este util pentru oamenii care vor sa isi repare masina, dar nu stiu la ce service. Astfel, ei vor afla experientele altor oameni la service-urile respective, pentru a lua cea mai buna decizie.

La fiecare service, orice client isi poate expune parerea despre serviciile oferite pentru ai ajuta pe urmatorii clienti sa ia cea mai decizie.

2) Un proprietar poate sa dea mai multe review-uri, 1 review la fiecare vizita in service. Un service are mai multe review-uri, de la fiecare client. Fiecare client da o nota de la 1 la 10 serviciilor primite la service-ul respectiv si mecanicului care a lucrat la masina lui. O masina poate sa fie reparata de mai multi mecanici si un mecanic poate sa repare mai multe masini. Intr-un judet exista mai multe service-uri. Pentru fiecare judet se face un top al service-urilor ce se va regasi in site-ul Top Service, unde se pot gasi si topurile din anii anteriori. Intr-un service au loc mai multe reparatii pentru diferite masini. Fiecare mecanic are un istoric unde sunt adunate review-urile de la clientii service-urile unde a lucrat.

3) Entitatile bazei de date sunt:

- Reviewers(PK: id_reviewer, U1:email) va avea date despre reviewer: nume, prenume, email, numar de telefon si id-ul masinii sale.
- Reviews(PK: id_review) va avea date despre review: id-ul celui care a facut review-ul, service-ul pentru care este review-ul, data review-ului si rating-ul dat de reviewer service-ului.
- Locations(PK:id_location) va avea date despre service: id-ul service-ului, id-ul judetului in care se afla service-ul, adresa si rating-ul service-ului.
- County(PK:id_county) va avea date despre judet: id-ul judetului si numele judetului.
- Top Service(PK:id_top_service) va avea date despre site-ul Top Service: id-ul paginii Top Service pentru un judet, id-ul judetului, anul aparitiei si id-ul locatiei.

- Top Service History(PK:id_county, year) va avea date despre pagina Top Service din fiecare an: id-ul service-ului, anul aparitiei, rating-ul service-ului si id-ul pagini Top Service din anul respectiv.
- Mechanics(PK:id_mechanic) va avea date despre mecanic: id-ul mecanic, numele, prenumele, id-ul service-ului unde lucreaza in prezent si rating-ul pe care l-a acumulat.
- Repair(PK:id_repair) va avea date depre reparatia unei masini: id-ul reparatiei, id-ul service-ului unde a avut loc reparatia si pretul reparatiei.
- Car repair(PK:id_car) va avea date despre masina: id-ul masini, id-ul service-ului, id-ul mecanicului, data inceperii reparatiei si data finalizarii reparatiei.
- Mechanic History(PK:id_mechanic, job_start) va avea date despre istoria mecanicului respectiv: id_mecanicului, data cand a inceput sa lucreze la service-ul respectiv si cand a terminat, id-ul service-ului si rating-ul acumulat.

4) Relatii:

- Reviewers-Reviews(1:m(mandatory))
- Reviewers-Car repair(1:m(optional))
- Mechanic-Car repair(m(mandatory):m(mandatory))
- Mechanics-Mechanic history(1:m(mandatory))
- Locations-Mechanics(1:m(mandatory))
- Locations-Repair(1:m(mandatory))
- County-Locations(1:m(mandatory))
- County-Top Service(1:1)
- Top Service-Top Service History(1:m(mandatory))

5) Atribute:

Reviewers:

- id_reviewer (int NOT NULL)
- first_name(varchar(25) NOT NULL)
- last_name(varchar(25) NOT NULL)
- email (varchar(50) NOT NULL)
- phone_number(int NOT NULL)
- id_car(int NOT NULL)

Reviews:

- id_review(int NOT NULL)
- id_reviewer(int NOT NULL)
- id_location(int NOT NULL)
- review_date(date NOT NULL)
- review_rating(int[from 1 to 10] NOT NULL)

County:

- id_county(int NOT NULL)
- county_name(varchar(35) NOT NULL)

Top Service:

- id_top_service(int NOT NULL)

- id_county(int NOT NULL)
- year(int NOT NULL)
- id_location(int NOT NULL)

Top Service History:

- id_county(int NOT NULL)
- year(int NOT NULL)
- review_rating(int [from 1 to 10] NOT NULL)
- id_top_service(int NOT NULL)

Locations:

- id_locations(int NOT NULL)
- id_county(int NOT NULL)
- address(varchar(50) NOT NULL)
- review_rating(int[from 1 to 10] NOT NULL)

Mechanics:

- id_mechanic(int NOT NULL)
- first_name(varchar(25) NOT NULL)
- last_name(varchar(25) NOT NULL)
- id_location(int NOT NULL)
- review_rating(int NOT NULL)

Repair:

- id_repair(int NOT NULL)
- id_location(int NOT NULL)
- price(int NOT NULL)

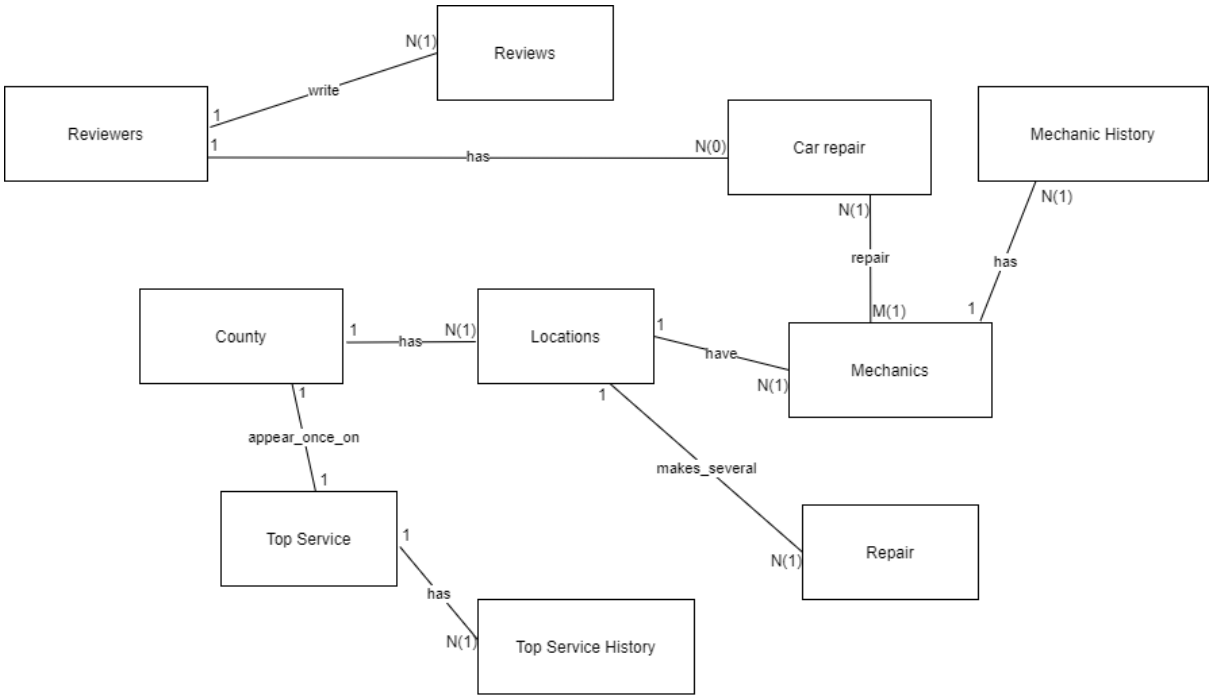
Car repair:

- id_car(int NOT NULL)
- id_location(int NOT NULL)
- id_mechanic(int NOT NULL)
- start_repair(date NOT NULL)
- end_repair(date NOT NULL)

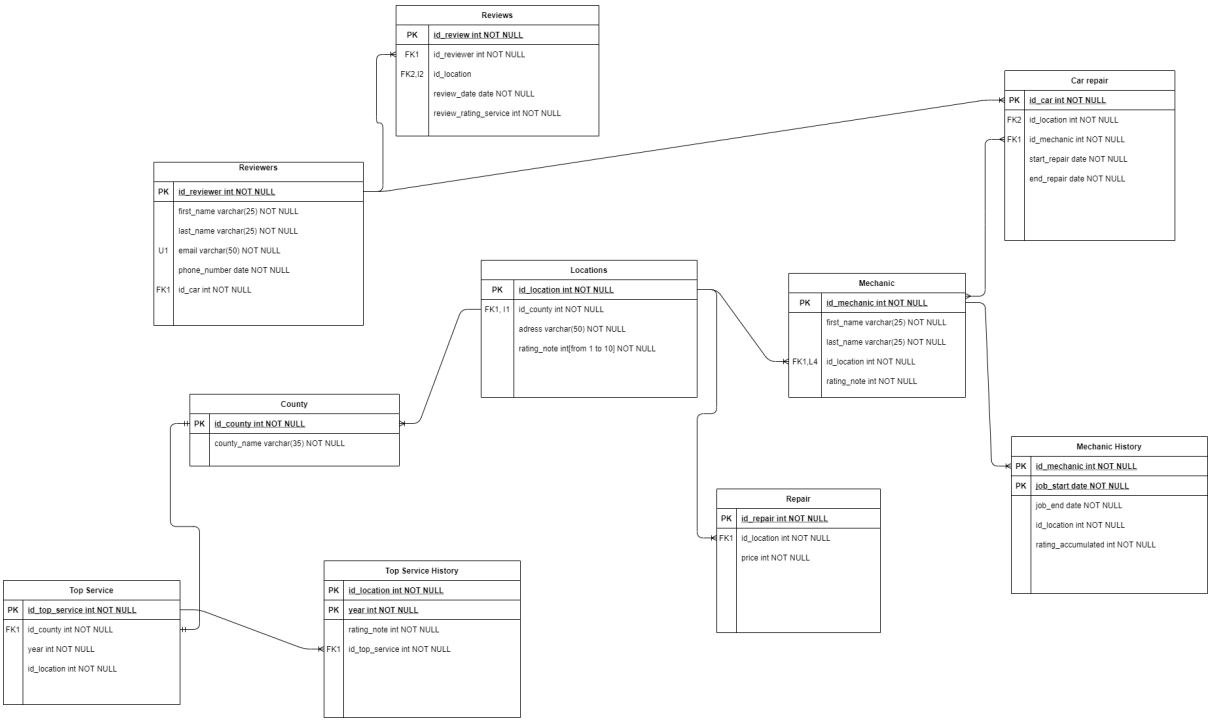
Mechanic History:

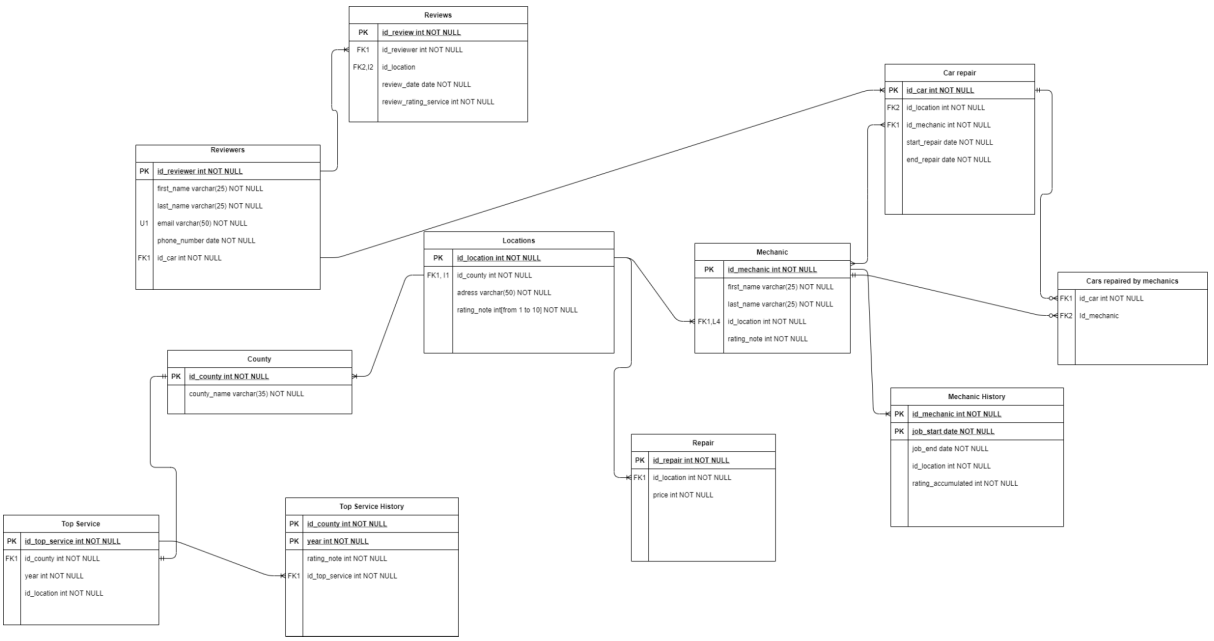
- id_mechanic(int NOT NULL)
- job_start(date NOT NULL)
- job_end(date NOT NULL)
- id_location(int NOT NULL)
- rating_accumulated(int NOT NULL)

6)ERD:



7) Diagrama conceptula:





8) Enumerarea schemelor relationale:

Reviewers:

- id_reviewer
- first_name
- last_name
- email
- phone_number
- id_car

Reviews:

- id_review
- id_reviewer
- id_location
- review_date
- review_rating

County:

- id_county
- county_name

Top Service:

- id_top_service
- id_county
- year
- id_location

Top Service History:

- id_county
- year
- review_rating
- id_top_service

Locations:

- id_locations
- id_county
- address
- review_rating

Mechanics:

- id_mechanic
- first_name
- last_name
- id_location
- review_rating

Repair:

- id_repair
- id_location
- price

Car repair:

- id_car
- id_location
- id_mechanic
- start_repair
- end_repair

Mechanic History:

- id_mechanic
- job_start
- job_end
- id_location
- rating_accumulated

Cars repaired by mechanics:

- id_car
- id_mechanic

9) Normalizarea pana la forma normala 3:

1)-Exeplu de baza de date NOT-NF1:

id_reviewer	first_name	last_name	email	phone_number
1	Marius	Manole	mariusmanile@yahoo.com mariusmanole@gmail.com	0789654334
2	Marius	Gicanu	mariusgicanu@yahoo.com	0786234567
3	Andrei	Tanase	andreitanase@yahoo.com	0736765234

Rezolvare NF1:

id_reviewer	first_name	last_name	last_name	phone_number
1	Marius	Manole	mariusmanole@yahoo.com	0789654334
1	Marius	Manole	mariusmanole@gmail.com	0789654334
2	Marius	Gicanu	mariusgicanu@yahoo.com	0786234567
3	Andrei	Tanase	andreitanase@yahoo.com	0736765234

2)-Exeplu de baza de date NF1 dar NOT-NF2:(exista dependente partiale)

id_mechanic	job_start	first_name	last_name	rating_note	rating_accumulated	id_location
1	05/05/2012	Tomi	Tomescu	5	5	1
1	06/11/2016	Tomi	Tomescu	7	13	2
3	01/01/2015	Radu	Iliescu	6	12	3

se transforma in:

id_mechanic	first_name	last_name	rating_note	id_location
1	06/11/2016	Tomi	Tomescu	2
2	01/01/2015	Radu	Iliescu	3

id_mechanic	first_name	last_name	rating_accumulated	job_start
1	Tomi	Tomescu	13	05/05/2012
3	Radu	Iliescu	12	01/01/2015

3)-Exeplu de baza de date NF2 dar NOT-NF3:(exista dependente tranzitive)

id_location	address	rating_note	id_county	county_name
1	str. Branduselor	8	1	Valcea
2	str. Pasarea	10	1	Valcea
3	str. Creanga	7	2	Brasov

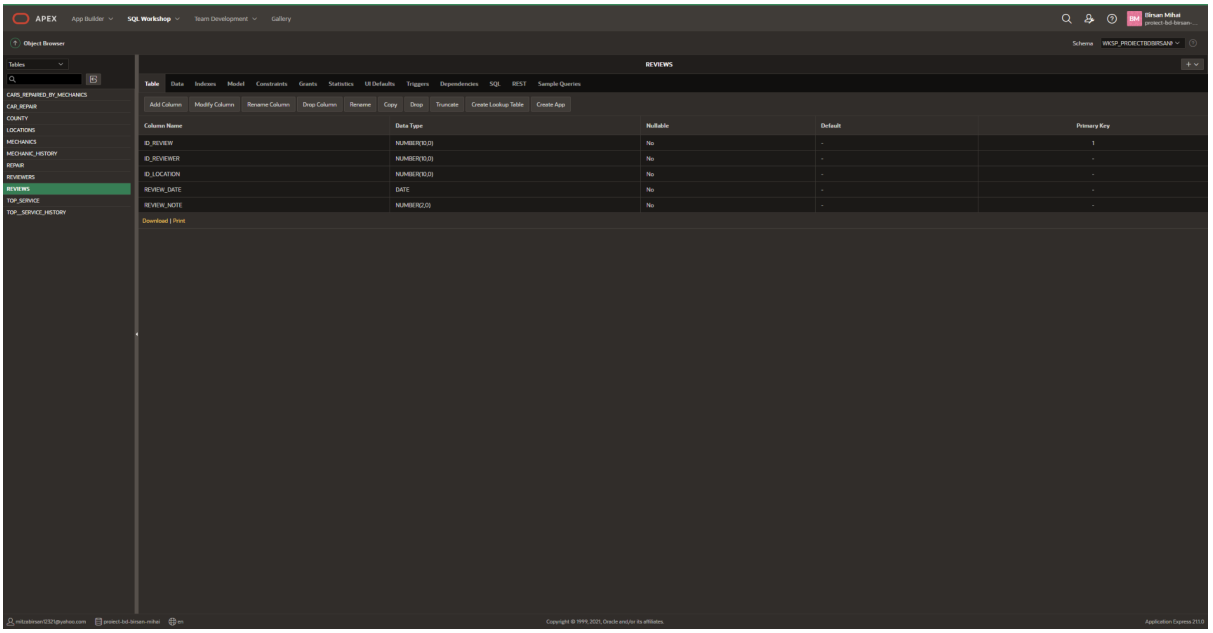
se transforma in:

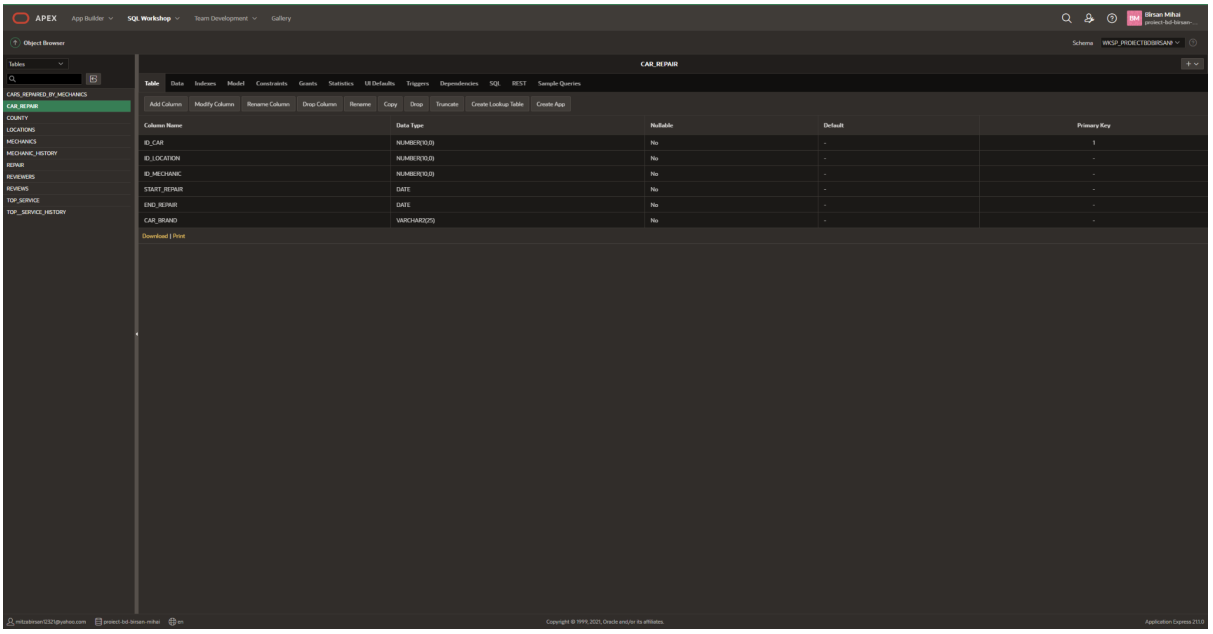
id_location	address	rating_note	id_county
1	str. Branduselor	8	1
2	str. Pasarea	10	1
3	str. Creanga	7	2

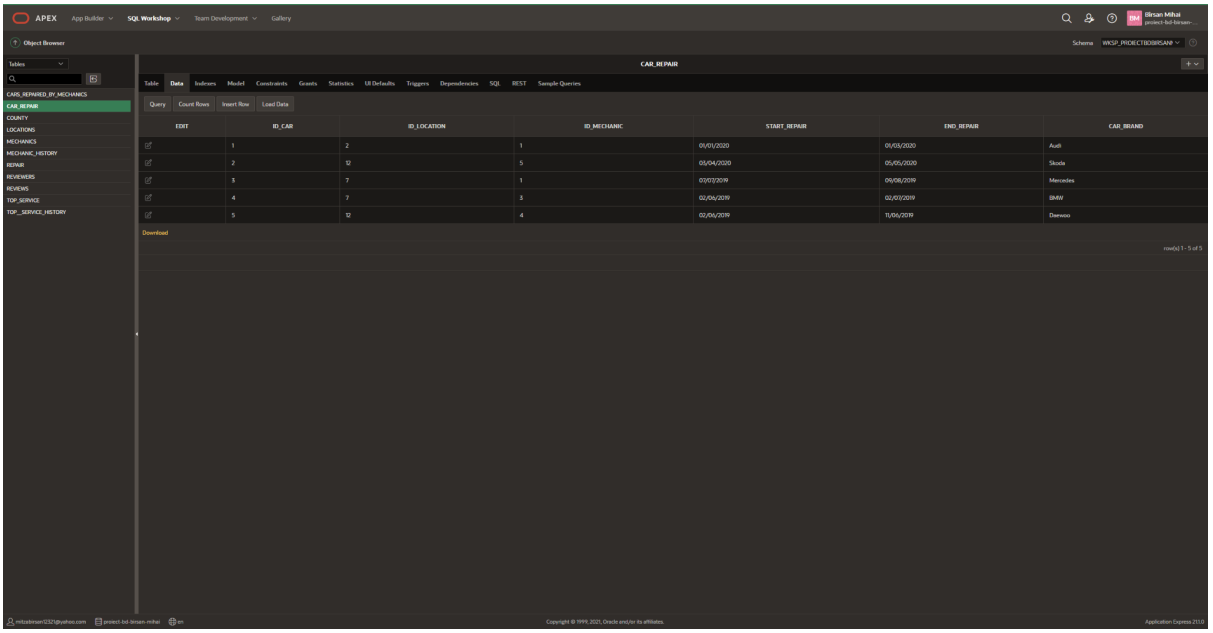
id_county	county_name
1	Valcea
2	Brasov

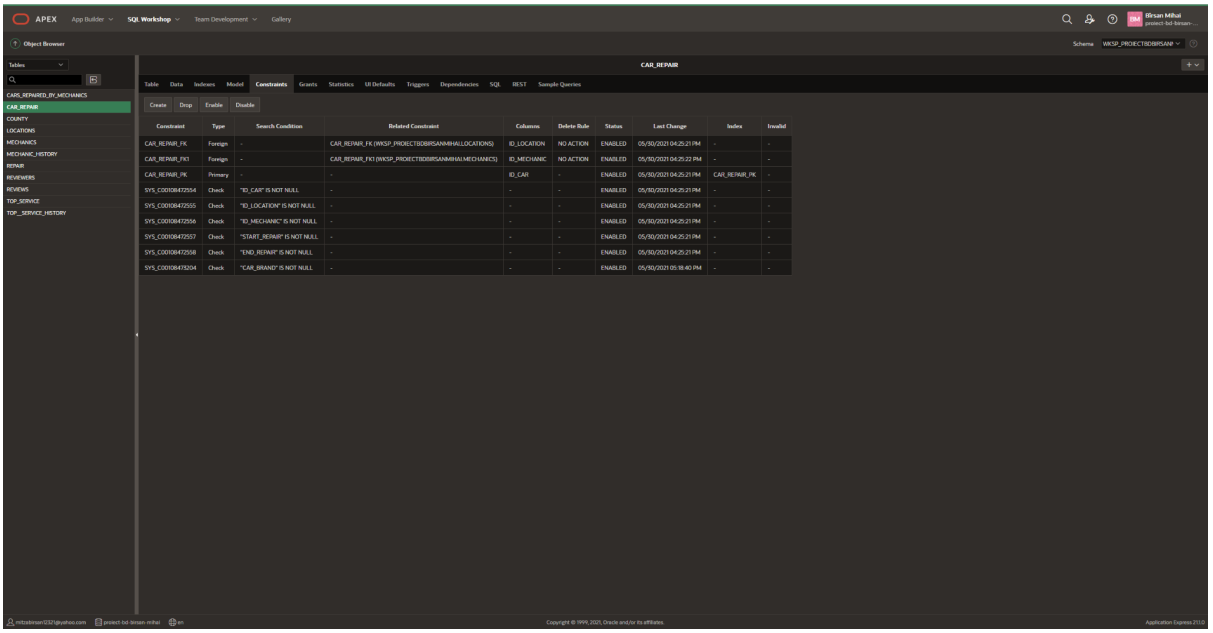
10)Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea:

```
CREATE TABLE  "MECHANIC_HISTORY"
   (    "ID_MECHANIC" NUMBER(10,0) NOT NULL ENABLE,
    "JOB_START" DATE NOT NULL ENABLE,
    "JOB_END" DATE,
    "ID_LOCATION" NUMBER(10,0) NOT NULL ENABLE,
    "RATING_ACCUMULATED" NUMBER(4,0) NOT NULL ENABLE,
     CONSTRAINT "MECHANIC_HISTORY_PK" PRIMARY KEY ("ID_MECHANIC",
"JOB_START")
  USING INDEX  ENABLE
   )
```






APEX
App Builder
SQL Workshop
Team Development
Gallery
Object Browser
Schema
WKSP_PROIECTBDBIRSANF
Tables
CARS_REPAIRED_BY_MECHANICS
CAR_REPAIR
COUNTY
LOCATIONS
MECHANICS
MECHANIC_HISTORY
REPAIR
REVIEWERS
REVIEWS
TOP_SERVICE
TOP_SERVICE_HISTORY
CAR_REPAIR
Table
Data
Indexes
Model
Constraints
Grants
Statistics
UI Defaults
Triggers
Dependencies
SQL
REST
Sample Queries
Query
Count Rows
Insert Row
Load Data
EDIT
ID_CAR
ID_LOCATION
ID_MECHANIC
START_REPAIR
END_REPAIR
CAR_BRAND
1 2 1 01/01/2020 01/03/2020 Audi
2 12 5 05/04/2020 05/05/2020 Skoda
3 7 1 07/07/2019 09/08/2019 Mercedes
4 7 3 02/06/2019 02/07/2019 BMW
5 12 4 02/06/2019 11/06/2019 Daewoo
Download
row(s) 1 - 5 of 5

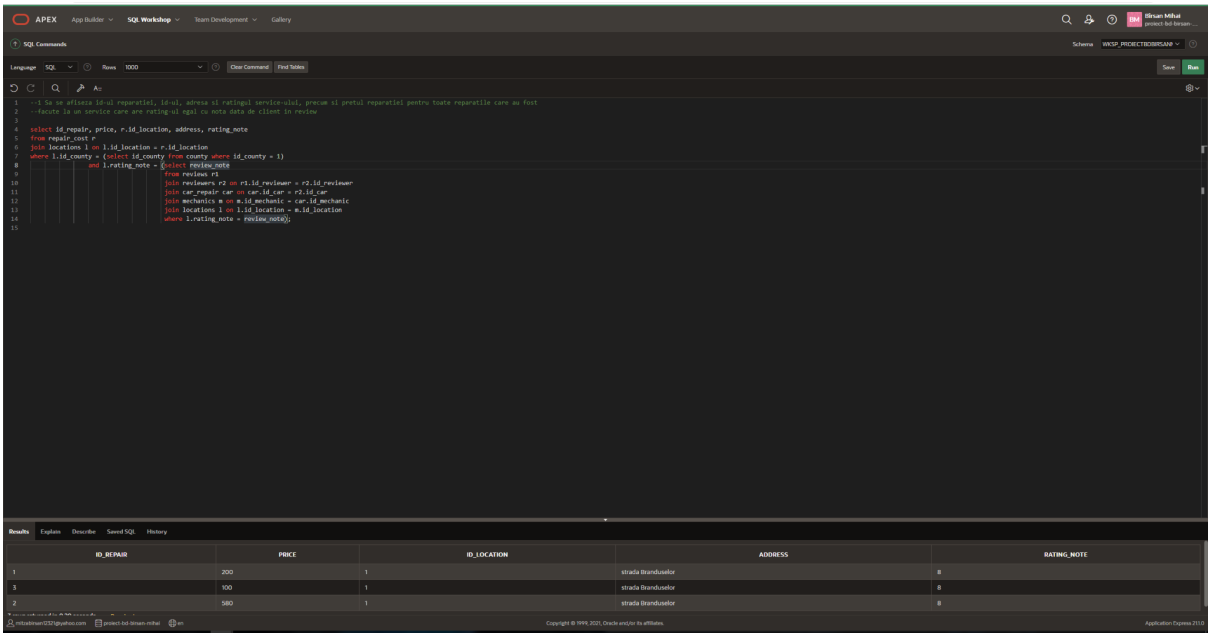

APEX
App Builder
SQL Workshop
Team Development
Gallery
Object Browser
Schema
WKSP_PROIECTBDBIRSANF
CAR_REPAIR
Table
Data
Indexes
Model
Constraints
Grants
Statistics
UI Defaults
Triggers
Dependencies
SQL
REST
Sample Queries
Create
Drop
Enable
Disable
Constraint
Type
Search Condition
Related Constraint
Columns
Delete Rule
Status
Last Change
Index
Invalid
CAR_REPAIR_FK Foreign - CAR_REPAIR_FK (WKSP_PROIECTBDBIRSANMIHAI.LOCATIONS) ID_LOCATION NO ACTION ENABLED 05/30/2021 04:25:21 PM - -
CAR_REPAIR_FK1 Foreign - CAR_REPAIR_FK1 (WKSP_PROIECTBDBIRSANMIHAI.MECHANICS) ID_MECHANIC NO ACTION ENABLED 05/30/2021 04:25:22 PM - -
CAR_REPAIR_PK Primary - - ID_CAR - ENABLED 05/30/2021 04:25:21 PM CAR_REPAIR_PK -
SYS_C00108472554 Check "ID_CAR" IS NOT NULL - - - ENABLED 05/30/2021 04:25:21 PM - -
SYS_C00108472555 Check "ID_LOCATION" IS NOT NULL - - - ENABLED 05/30/2021 04:25:21 PM - -
SYS_C00108472556 Check "ID_MECHANIC" IS NOT NULL - - - ENABLED 05/30/2021 04:25:21 PM - -
SYS_C00108472557 Check "START_REPAIR" IS NOT NULL - - - ENABLED 05/30/2021 04:25:21 PM - -
SYS_C00108472558 Check "END_REPAIR" IS NOT NULL - - - ENABLED 05/30/2021 04:25:21 PM - -
SYS_C00108473204 Check "CAR_BRAND" IS NOT NULL - - - ENABLED 05/30/2021 05:18:40 PM - -

11) 5 cereri SQL:

a)

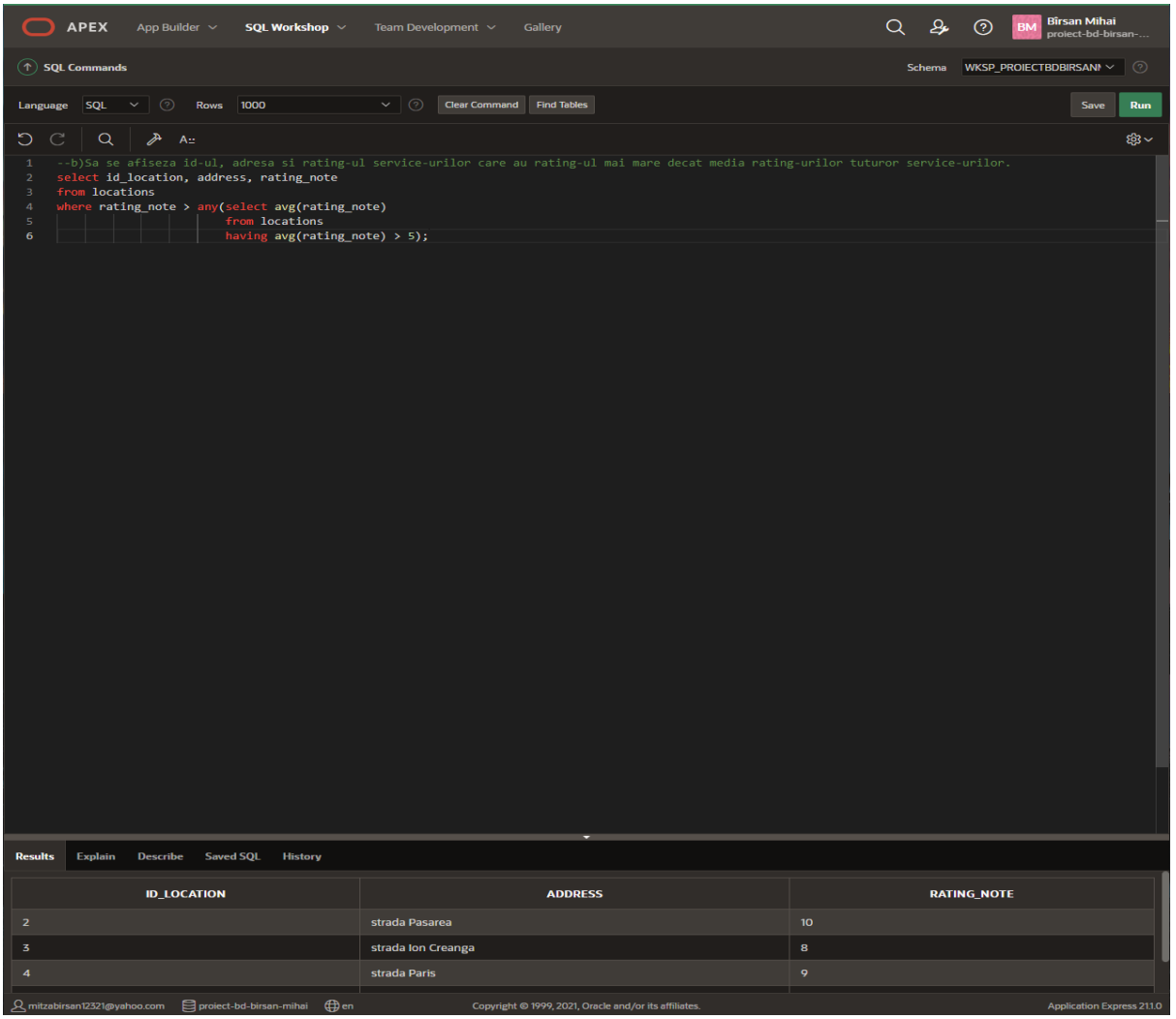
```
      --1 Sa se afiseza id-ul reparatiei, id-ul, adresa si ratingul
service-ului, precum si pretul reparatiei pentru toate reparatile care
au fost
      --facute la un service care are rating-ul egal cu nota data de
client in review

      select id_repair, price, r.id_location, address, rating_note
      from repair_cost r
      join locations l on l.id_location = r.id_location
      where l.id_county = (select id_county from county where id_county
= 1)
                        and l.rating_note = (select review_note
                                             from reviews r1
                                             join reviewers r2 on
r1.id_reviewer = r2.id_reviewer
                                             join car_repair car on
car.id_car = r2.id_car
                                             join mechanics m on
m.id_mechanic = car.id_mechanic
                                             join locations l on
l.id_location = m.id_location
                                             where l.rating_note =
review_note);
```



b)

```
    --b)Sa se afiseza id-ul, adresa si rating-ul service-urilor care
au rating-ul mai mare decat media rating-urilor tuturor service-urilor.
    select id_location, address, rating_note
    from locations
    where rating_note > any(select avg(rating_note)
                            from locations
                            having avg(rating_note) > 5)
    order by id_location;
```



c)

```
--3 Sa se afiseze prenumele, numele,id-ul service-ului unde
lucreaza, id-ul mecanicilor care au rating-ul maxim si care lucreza la
service-ul cu id-ul 2.

with best_mechanic as (
    select first_name, last_name, id_mechanic, id_location
    from mechanics
    where rating_note = (select max(rating_note)
                         from mechanics ))
select *
from best_mechanic
where id_location = 2;
```

FIRST_NAME	LAST_NAME	ID_MECHANIC	ID_LOCATION
Roger	Filipescu	7	2
Valeriu	Blescu	6	2

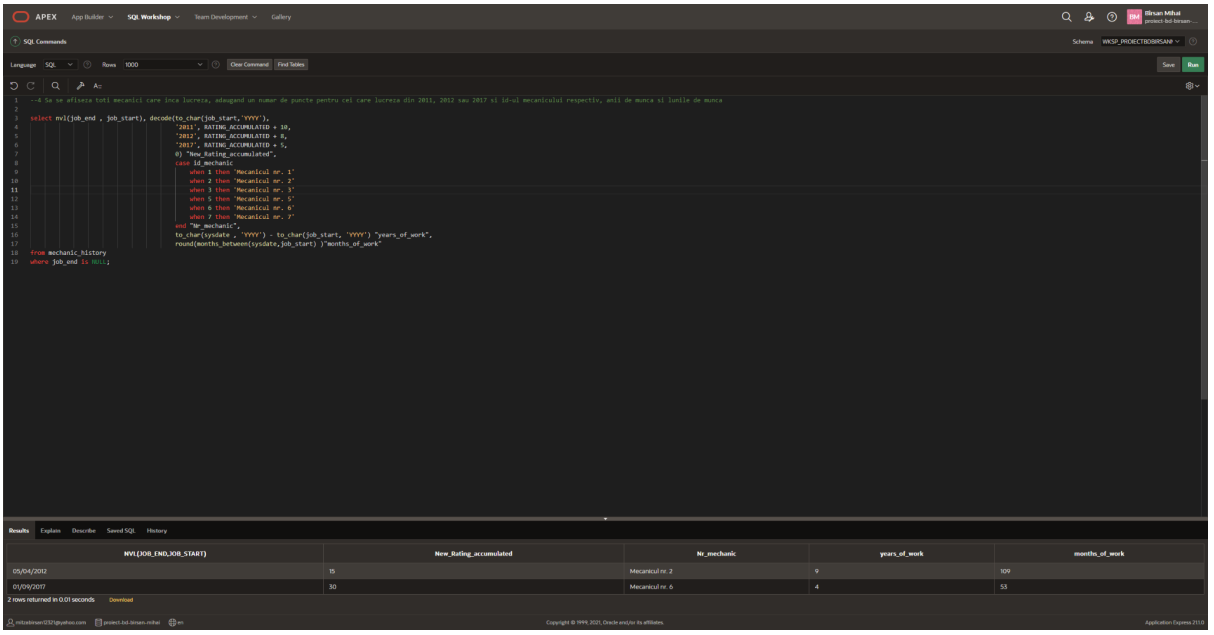
d)

```
--4 Sa se afiseza toti mecanici care inca lucreza, adaugand un
numar de puncte pentru cei care lucreza din 2011, 2012 sau 2017 si
id-ul mecanicului respectiv, anii de munca si lunile de munca

select nvl(job_end , job_start),
decode(to_char(job_start,'YYYY'),
                                        '2011',
RATING_ACCUMULATED + 10,
                                        '2012',
RATING_ACCUMULATED + 8,
                                        '2017',
RATING_ACCUMULATED + 5,
```

```
                                                0)
"New_Rating_accumulated",
                                                case id_mechanic
                                                    when 1 then
'Mecanicul nr. 1'
                                                    when 2 then
'Mecanicul nr. 2'
                                                    when 3 then
'Mecanicul nr. 3'
                                                    when 5 then
'Mecanicul nr. 5'
                                                    when 6 then
'Mecanicul nr. 6'
                                                    when 7 then
'Mecanicul nr. 7'
                                                end "Nr_mechanic",
                                                to_char(sysdate , 'YYYY')
- to_char(job_start, 'YYYY') "years_of_work",

round(months_between(sysdate,job_start) )"months_of_work"
      from mechanic_history
      where job_end is NULL;
```



e)

```
      --sa se afiseza numele, prenumele, rating-ul si id-ul
service-ului unde lucreaza mecanicul care are rating-ul mai mare decat
media rating-urile realizata
      --din review-uri pe care le-au dat clientii respectivului
service.

      select first_name, last_name, rating_note, id_location
      from mechanics cel_afis
      where rating_note > (select avg(rating_note)
                           from reviews r1
                           join reviewers r2 on r1.id_reviewer =
r2.id_reviewer
                           join car_repair car on car.id_car =
r2.id_car
                           join mechanics m on m.id_mechanic =
car.id_mechanic
                           where m.id_location = cel_afis.id_location);
```

12)Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri:

```
UPDATE repair_cost
SET id_location = 3, price = 1000
WHERE id_location = (select id_location
                     from mechanics
                     where lower(first_name) = 'tomi');
```


```
UPDATE reviews
SET review_note = 9, id_location = 1
WHERE id_reviewer = (select id_reviewer
                     from reviewers
                     where id_car = 5);
```


```
UPDATE top_service
SET id_location = 1
WHERE id_location = (select id_location
                     from mechanics
                     where rating_note = 8);
```

13). Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele:

```
INSERT INTO car_repair (id_car, id_location, id_mechanic,
start_repair, end_repair, car_brand)
VALUES (..., ..., ..., ..., ..., ...);
```

```
      INSERT INTO county (id_county, county_name)
      VALUES (..., ...);

      INSERT INTO locations (id_locations, id_county, address,
rating_note)
      VALUES (..., ..., ..., ...);

      INSERT INTO mechanics (id_mechanic, first_name, last_name,
id_location, rating_note)
      VALUES (..., ..., ..., ..., ...);

      INSERT INTO mechanic_history (id_mechanic, job_start, job_end,
id_location, rating_accumulated)
      VALUES (..., ..., ..., ..., ...);

      INSERT INTO repair_cost (id_repair, id_location, price)
      VALUES (..., ..., ...);

      INSERT INTO reviewers (id_reviewer, first_name, last_name, email,
phone_number, id_car)
      VALUES (..., ..., ..., ..., ..., ...);

      INSERT INTO reviews (id_review, id_reviewer, id_location,
review_date, review_note)
      VALUES (..., ..., ..., ..., ...);

      INSERT INTO top_service (id_top_service, id_county, year,
id_location)
      VALUES (..., ..., ..., ...,);

      INSERT INTO top_service_history (id_county, year, rating_note,
id_top_service)
      VALUES (..., ..., ..., ...);
```

14) Crearea unei vizualizări compuse:

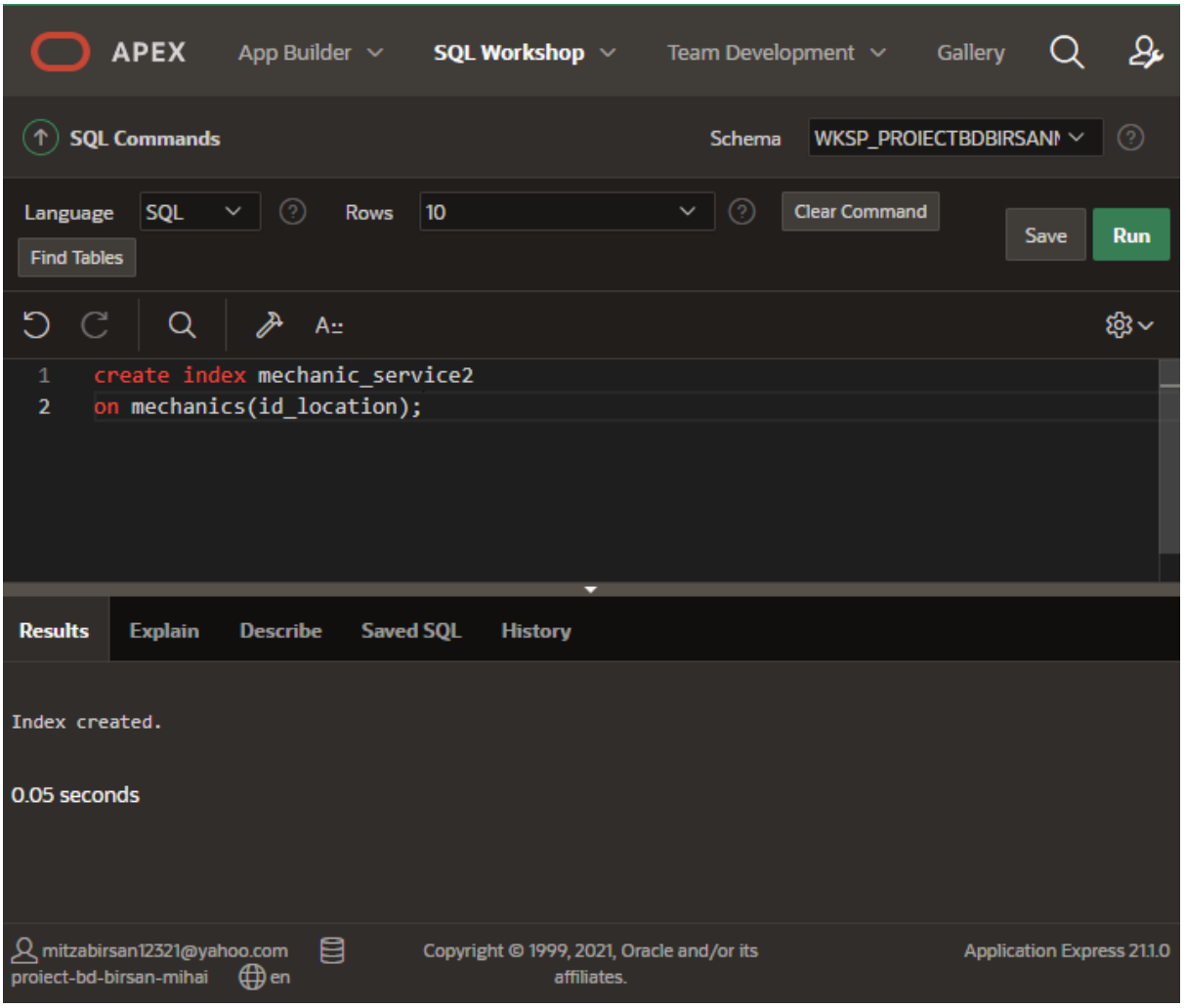
```
      --un view pentru service-urile din Valcea
      create view Locations_in_Valcea
      as select id_location, l.id_county, address, rating_note,
county_name
          from locations l
          join county co on l.id_county = co.id_county
          where lower(county_name) = 'valcea';
```

Operatii LMD nepermise:

1. funcţii grup,
2. clauzele GROUP BY, HAVING ,START WITH , CONNECT BY
3. cuvântul cheie DISTINCT
4. pseudocoloana ROWNUM
5. operatori pe mulţimi.

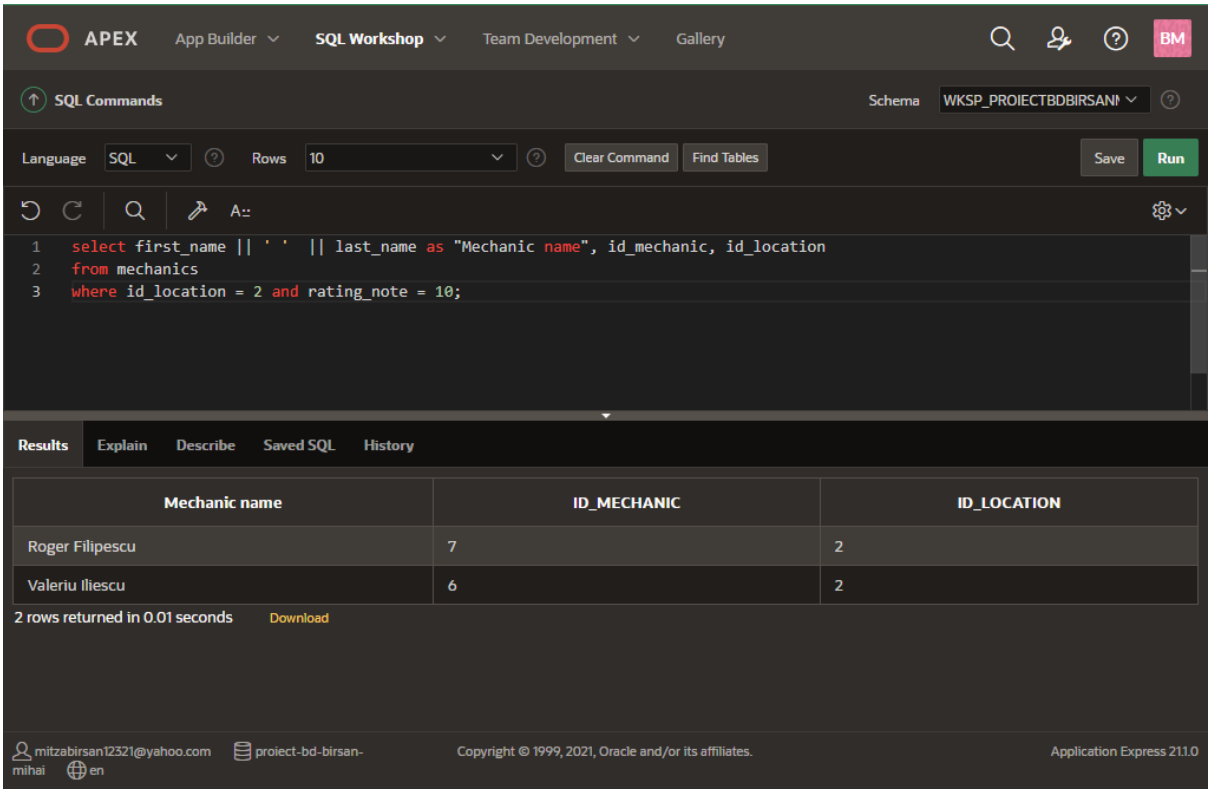
15)

```
create index mechanic_service2
on mechanics(id_location);
```



Astfel optimizeaza operatile de genul:

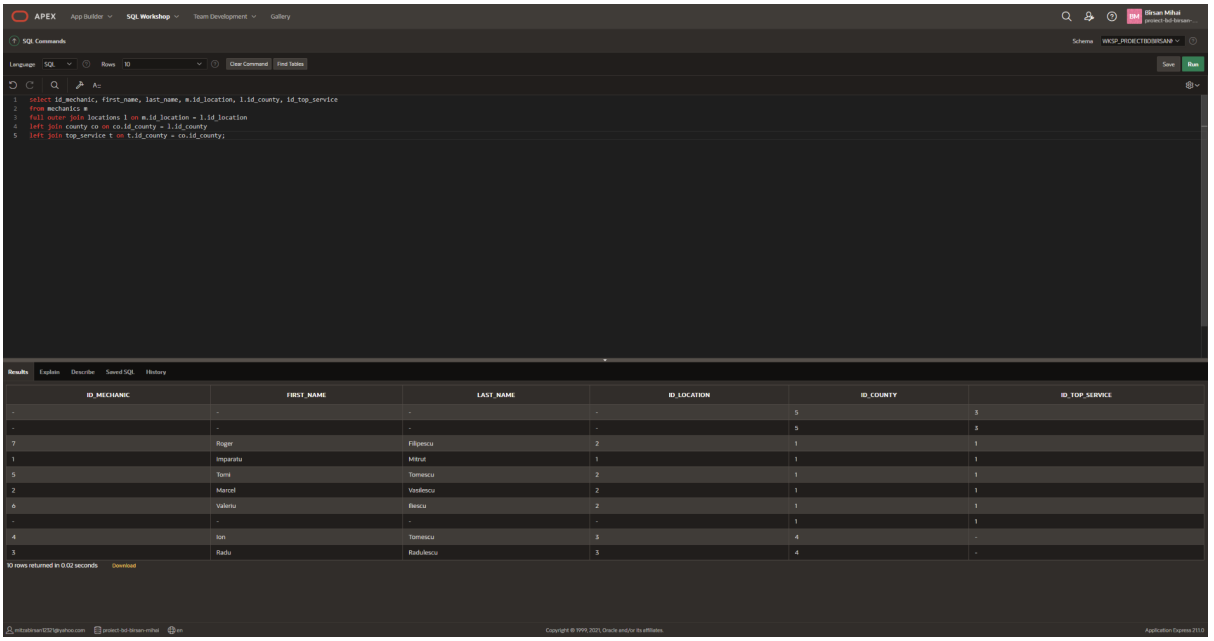
```
      select first_name || ' '  || last_name as "Mechanic name",
id_mechanic, id_location
      from mechanics
      where id_location = 2 and rating_note = 10;
```



16)

Sa se afiseze din toate judetele: toate service-urile, toti mecanici de la toate service-urile si toate aparitile service-urilor din pagina Top Service.

```
      select id_mechanic, first_name, last_name, m.id_location,
l.id_county, id_top_service
      from mechanics m
      full outer join locations l on m.id_location = l.id_location
      left join county co on co.id_county = l.id_county
      left join top_service t on t.id_county = co.id_county;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
SQL Commands
Language SQL
Rows 10
Clear Command
Find Tables
Save
Run
1 select id_mechanic, first_name, last_name, m.id_location, l.id_county, id_top_service
2 from mechanics m
3 full outer join locations l on m.id_location = l.id_location
4 left join county co on co.id_county = l.id_county
5 left join top_service t on t.id_county = co.id_county;
Results
Explain
Describe
Saved SQL
History
ID_MECHANIC FIRST_NAME LAST_NAME ID_LOCATION ID_COUNTY ID_TOP_SERVICE
- - - - 5 3
- - - - 5 3
7 Roger Filipescu 2 1 1
1 Imparatu Mitrut 1 1 1
5 Tomi Tomescu 2 1 1
2 Marcel Vasilescu 2 1 1
6 Valeriu Bescu 2 1 1
- - - - 1 1
4 Ion Tomescu 3 4 -
3 Radu Radulescu 3 4 -
10 rows returned in 0.02 seconds
Download
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.0