

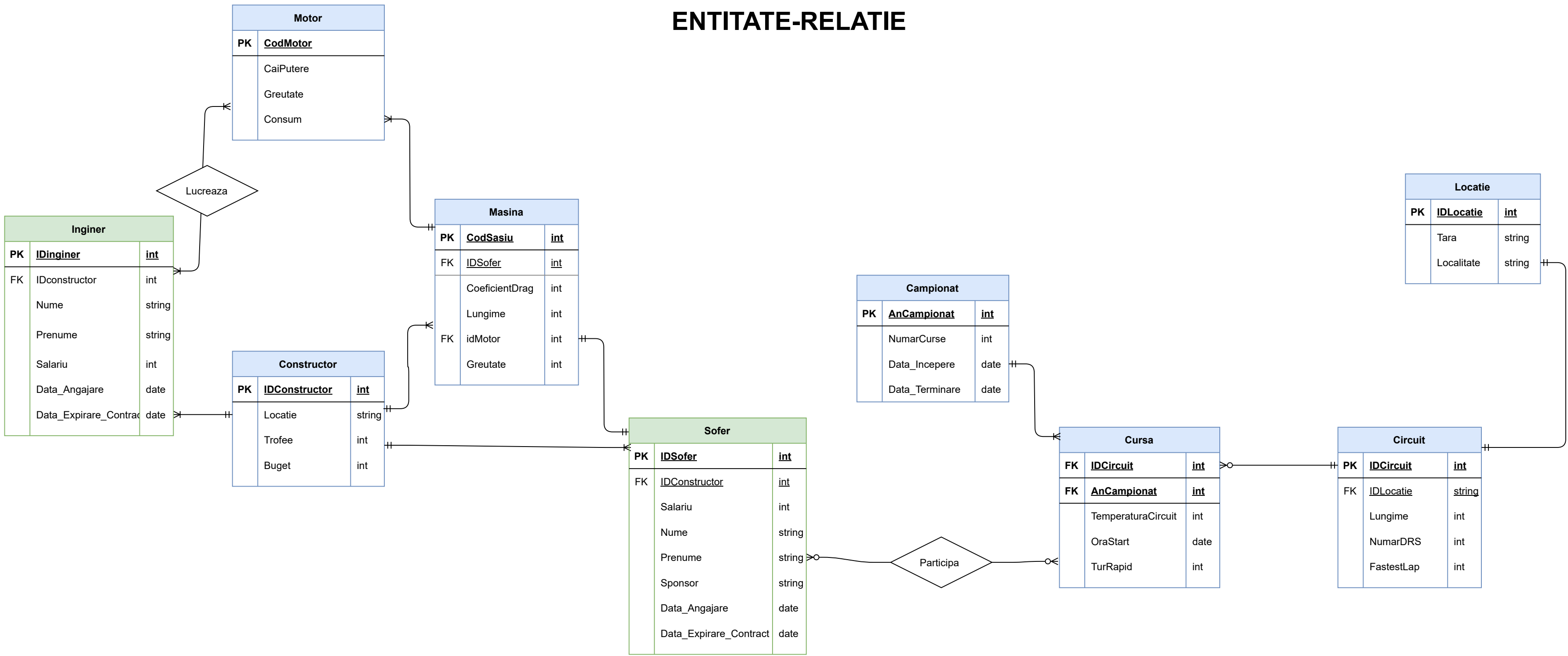
Schema relatie campionat formula 1: O competitie de formula 1 este formata din 10 constructori, un constructor are 2 soferi , soferii nu pot schimba masini intre ei. Un an competitional ofera 23 curse, aflate pe circuite diferite. Pozitia pe care un sofer incepe cursa este data de calificarea pe care ei o fac cu o zi inainte de cursa, pe acelasi circuit.

Constructorul, de asemenea, are ingineri care se ocupa de dezvoltarea masinii.

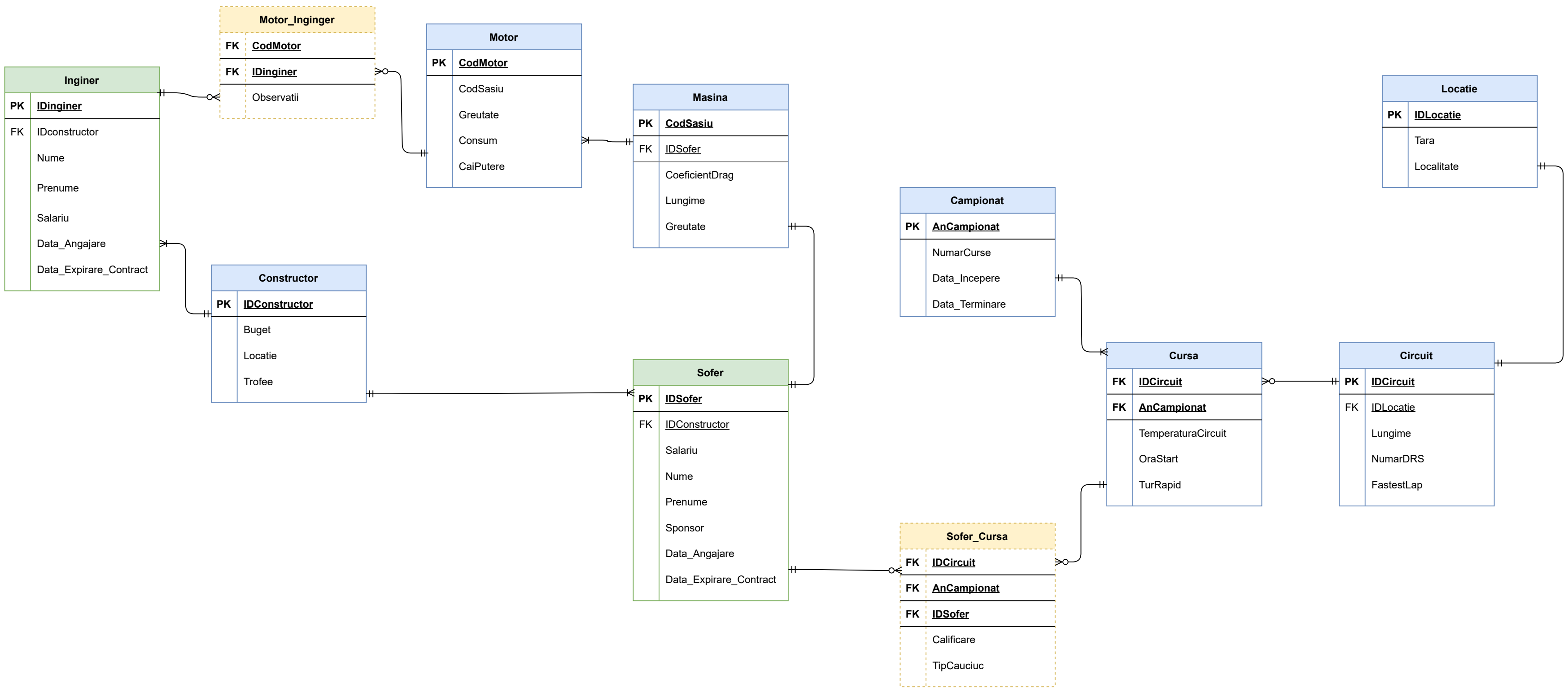
Campionatul se desfasoara de-a lungul unui an (ex campionatul din anul 2015).

Unicitatea unei curse este determinata de circuitul pe care este facuta si anul acesteia. (O cursa nu poate fi de doua ori in acelasi an pe acelasi circuit).

ENTITATE-RELATIE



CONCEPTUALA



Cerinta 8:

Ingerin(ID_Ingerin,Nume, Prenume, Salariu, Data_Angajare,Data_Expirare_Contract),
Constructor(ID_Constructor, Buget, Locatie, Trofee),
Masina(Cod_Sasiu, CoeficientDrag, Lungime, Putere, Greutate),
Sofer(ID_Sofer, Salariu, Nume, Prenume, Sponsor, Data_Angajare,Data_Expirare_Contract),
Sofer_Cursa(Calificare, TipCauciuc)
Campionat(AnCampionat, NumarCursa, Data_Incepere, Data_Terminare),
Cursa(ID_Circuit, AnCampionat, TemperaturaCircuit, OraStart, TurRapid),
Circuit(ID_Circuit, Lungime, NumarDRS, FastestLap),
Locatie(ID_Locatie, Tara, Localitate)

Cerinta 9

NON-FN1

Tara	Localitate
Italia	Monza, Benevento
Ungaria	Budapesta
Franta	Paris, Marseille

FN1

IDLocatie	Tara	Localitate
1	Italia	Monza
2	Italia	Benevento
3	Ungaria	Budapesta
4	Franta	Paris
5	Franta	Marseille

NON-FN2

FN2

IDCircuit	AnCampionat	TempCircuit	OraStart	IDLocatie	Lungime	NumarDRS	FastestLap
1	2018	30	13:00	1	5.08km	2	1:30.888
1	2019	28	14:00	1	5.08km	2	1:30.888
2	2019	31	15:00	2	5.12km	3	1:42.120
3	2019	33	16:00	3	5.77km	1	1:51.230

IDCircuit	AnCampionat	TempCircuit	OraStart
1	2018	30	13:00
1	2019	28	14:00
2	2019	31	15:00
3	2019	33	16:00

DEPENDENTE

IDCIRCUIT->LUNGIME+NUMARDRS+FASTESTLAP

IDCIRCUIT-+ANCAMPIONAT->TEMPCIRCUIT,ORASTART

IDCircuit	IDLocatie	Lungime	NumarDRS	FastestLap
1	1	5.08km	2	1:30.888
2	2	5.12km	3	1:42.120
3	3	5.77km	1	1:51.230

NON-FN3

FN3

Codmotor	IDInghner	Observati	CodSasu	GreutateM	Consum	Caiputere	CofeDrac	Lungtime	GreutateMas
1	1	Bun	202	300	23.4	880	0.03	5.8	560
2	1	Ok	203	280	24.2	1002	0.05	5.12	610
3	2	Ok	203	255	24.2	1002	0.05	5.8	610
4	3	FB	205	288	20.1	1129	0.14	5	588

Codmotor	IDInginer	Observatii
1	1	Bun
2	1	Ok
3	2	Ok
4	3	FB

DEPENDENTE

CODMOTOR+IDINGINER ->OBSERVATI

CODMOTOR->CODSASIU,GREUTATEM,CONSUM,CAIPUTERE

CODSASIU->COEFDRA, LUNGIME, GREUTATE MASINA

Codmotor	CodSasiu	GreutateM	Consum	Calputere	CoefDrag	Lungime	GreutateMas
1	202	300	23.4	880	0.03	5.8	560
2	203	280	24.2	1002	0.05	5.12	610
3	203	255	24.2	1002	0.05	5.12	610
4	205	288	20.1	1129	0.14	5	588

Codmotor	CodSasiu	GreutateM	Consum	Caiputere
1	202	300	23.4	880
2	203	280	24.2	1002
3	203	255	24.2	1002
4	205	288	20.1	1129

CodSasiu	CoefDrag	Lungime	GreutateMas
202	0.03	5.8	560
203	0.05	5.12	610
205	0.14	5	588

Screenshot cu rularea tuturor comenzilor

Cerinta11

```
105
106 • insert into Vizualizare (idinginer, nume, prenume, salariu, data_angajare, data_expirare, idconstructor, NumConstructor) values (8, 'Pipsick', 'Daniel', 8000, '2018-04-20', '2021-05-21', 1, 'Renault');
107
108 • rollback;
109 • select * from Vizualizare;
110
111 /* CERINTA 15 */
112
113 • create index full_name
114   on inginer (nume, prenume);
115
116 • show index from inginer;
117
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	codsasiu	idsofer	codmotor	coefficientdrag	greutate
▶ 202	1	1	0	722	
203	2	1	0	704	
204	3	2	0	722	
205	4	3	0	733	
206	5	4	0	724	

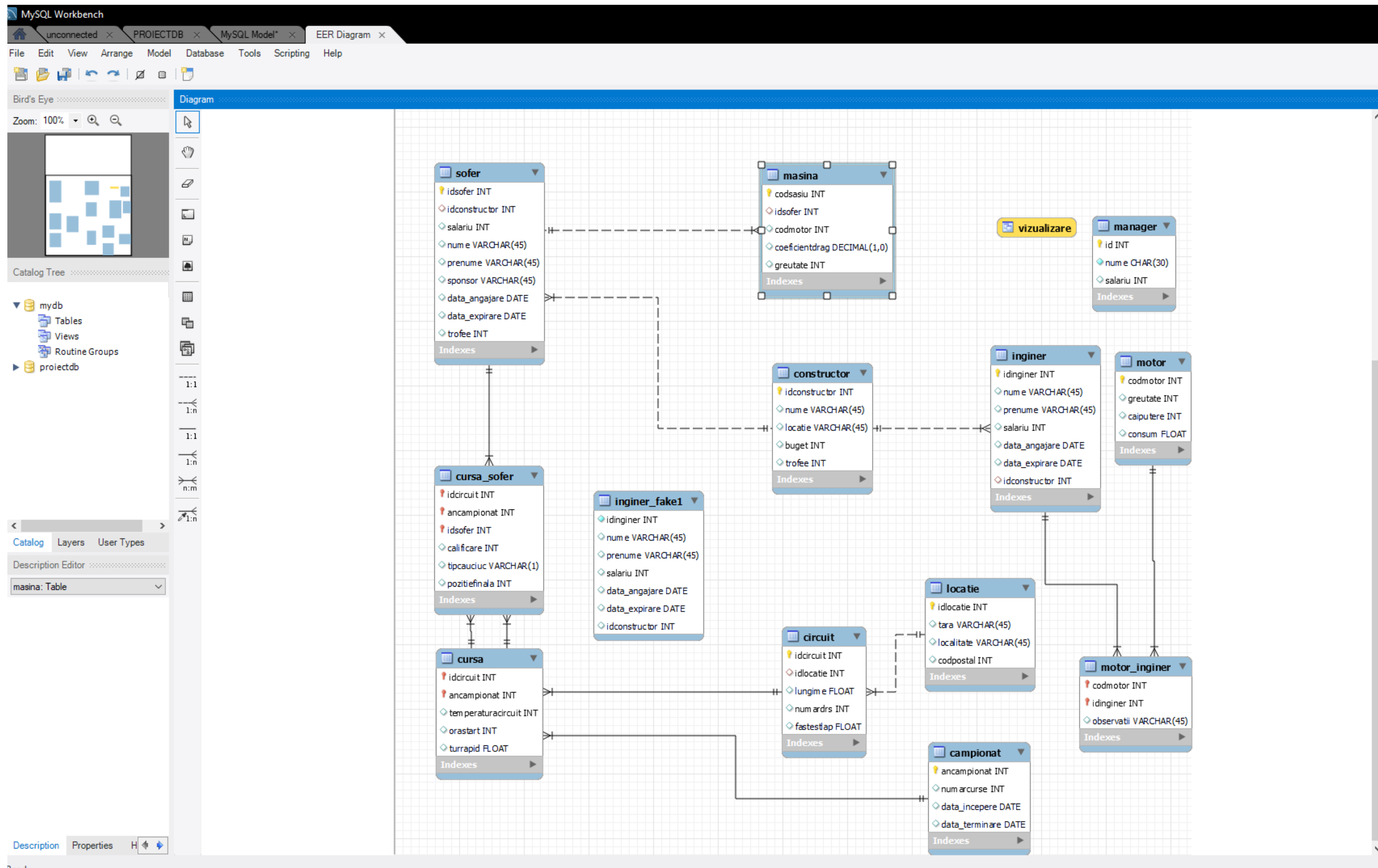
Result 7 Result 8 Result 9 Result 10 Result 11 sofer 12 inginer 13 masina 14 x Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 12	10:47:08	ROLLBACK	0 row(s) affected	0.000 sec
✓ 13	10:47:08	ROLLBACK	0 row(s) affected	0.000 sec
✓ 14	10:47:16	DROP TABLE `proiectdb`.`inginer_fake1`	0 row(s) affected	0.031 sec
✓ 15	10:47:21	select c.idcircuit,p.idsofer,p.pozitiefinala,d.data_incepere from cursa c join campionat d on(c.ancampionat = d.ancampionat) join circuit e on (c.idcircuit...	2 row(s) returned	0.000 sec / 0.000 sec
✓ 16	10:47:21	select ifnull(count(idcircuit), 'Nu are') as NumarCircuite, ifnull(c.ancampionat, 'Nu are') as AnCampionat from cursa c right outer join campionat d on (c.a...	5 row(s) returned	0.000 sec / 0.000 sec
✓ 17	10:47:21	WITH inginer AS (SELECT * FROM inginer where salariu > 9900) SELECT * FROM inginer WHERE month(data_angajare) = 8 UNION ALL SELECT * ...	1 row(s) returned	0.000 sec / 0.000 sec
✓ 18	10:47:21	select s.numa, s.prenume, s.salariu, case when s.idconstructor = 1 then 'Mercedes' when s.idconstructor =2 then 'Red bull' when s.idconstructor =3...	1 row(s) returned	0.000 sec / 0.000 sec
✓ 19	10:47:21	select s.numa, s.prenume, s.salariu, s.idconstructor, b.trofee from sofer s join constructor b on (s.idconstructor = b.idconstructor) where s.idsofer not in(...	1 row(s) returned	0.000 sec / 0.000 sec
✓ 20	10:47:21	UPDATE sofer set trofee = (select count(pozitiefinala) from cursa_sofer c where pozitiefinala= 1 and c.idsofer = sofer.idsofer)	2 row(s) affected Rows matched: 7 Changed: 2 Warnings: 0	0.000 sec
✓ 21	10:47:21	rollback	0 row(s) affected	0.000 sec
✓ 22	10:47:21	select * from sofer LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
✓ 23	10:47:21	CREATE TABLE inginer_fake1 AS SELECT * FROM inginer	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.031 sec
✓ 24	10:47:21	update inginer set salariu = (select avg(salariu) from inginer_fake1) where inginer.salariu < 10000	5 row(s) affected Rows matched: 5 Changed: 5 Warnings: 0	0.000 sec
✓ 25	10:47:21	rollback	0 row(s) affected	0.000 sec
✓ 26	10:47:21	select * from inginer LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
✓ 27	10:47:21	delete from masina where idsofer = any(select idsofer from sofer where sponsor = "Merc")	1 row(s) affected	0.000 sec
✓ 28	10:47:21	rollback	0 row(s) affected	0.015 sec
✓ 29	10:47:21	select * from masina LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Screenshot cu diagrama bazei de date, generata in mysql workbench



Cerinta 11:

```
/*1. Join cu 4 tabele, folosind where si functie cu data. Acesta selecteaza idcircuit, sofer, pozitiefinala si data incepere pentru soferii
care au terminat pe primul loc intr-o cursa care a avut loc intr-un campionat care a inceput cu cel putin un an mai devreme decat data curenta.*/

select c.idcircuit,p.idsofer,p.pozitiefinala,d.data_incepere
from cursa c join campionat d on(c.ancampionat = d.ancampionat)
join circuit e on (c.idcircuit = e.idcircuit)
join cursa_sofer p on (p.idcircuit = c.idcircuit)
where pozitiefinala = 1 and datediff(current_date(),data_incepere)>365;

/*2. Selecteaza numarul de campionate care au avut loc intr-un an folosind functia group by, ordonata folosind functia order by, descendent, in functie de numarul de circuite, in care anul campionatului e mai mic decat 2018
Am folosit functia ifnull, sql nu are NVL, se numeste ifnull*/

select ifnull(count(idcircuit), 'Nu are') as NumarCircuite, ifnull(c.ancampionat, 'Nu are') as AnCampionat
from cursa c right outer join campionat d on (c.ancampionat = d.ancampionat)
group by c.ancampionat
having count(idcircuit)<=1
order by count(idcircuit) DESC;

/*3. Face o reuniune dintre inginerii care au numele "Gica" si cei care au mai mult de 2880 de zile de la data angajarii, folosind clauzele with si union*/
WITH inginer AS (SELECT * FROM inginer where salariu > 9999)
SELECT * FROM inginer WHERE month(data_angajare) = 8
UNION ALL
SELECT * FROM inginer WHERE nume = 'Gica'
UNION ALL
select * from inginer where prenume = 'Dorel';

/*4. Cerere sincronizata unde selectam numele, prenume, salariul, idconstructor si trofeele acestuia , pentru soferii care nu au participat la nicio cursa */

select s.num, s.prenume, s.salariu,
case when s.idconstructor = 1 then 'Mercedes'
when s.idconstructor =2 then 'Red bull'
when s.idconstructor =3 then 'Ferrari'
when s.idconstructor = 4 then 'McLaren'
else 'Necunoscut'
end as numeConstructor,
b.trofee from sofer s join constructor b on (s.idconstructor = b.idconstructor)
where not exists(select idsofer from cursa_sofer where idsofer = s.idsofer );

/* 5.Aceiasi cerere de mai sus, numai ca nesincronizata.*/
select s.num, s.prenume, s.salariu, s.idconstructor, b.trofee from sofer s join constructor b on (s.idconstructor = b.idconstructor)
where s.idsofer not in(select idsofer from cursa_sofer where idsofer is not null );
```

Cerinta 12+13:

- ```
/* CERINTA 12*/
/* 1. Update care ii atribuie numarul de trofee unui sofer, doar daca acesta a luat locul 1 la o cursa*/

UPDATE sofer
set trofee = (select count(pozitiefinala) from cursa_sofer c where pozitiefinala= 1 and c.idsofer = sofer.idsofer);

rollback;

select * from sofer;

/* 2.*/
/*Facem un tabel nou, iar aceasta instructiune le atribuie tuturor inginerilor cu salariu mai mic decat 10000, salariul mediu)*/

CREATE TABLE inginer_fake1 AS SELECT * FROM inginer;

update inginer
set salariu = (select avg(salariu) from inginer_fake1)
where inginer.salariu < 10000;

rollback;

select * from inginer;

/* 3. */
/*Sterge liniile masinilor al caror sofer este sponsorizat de firma "merc"*/

delete from masina
where idsofer = any(select idsofer from sofer where sponsor = "Merc");

rollback;

select * from masina;

/*CERINTA 13 -----MYSQL NU FOLOSESTE SEQUENCES, ASA CA O SA MA FOLOSESC DE FUNCTIA AUTO-INCREMENT*/

/*Aceasta creaza un tabel numit manager, unde cheia primara, id-ul, se incrementeaza cu 1 la fiecare insert*/

CREATE TABLE manager (
 id int NOT NULL PRIMARY KEY AUTO_INCREMENT,
 nume CHAR(30) NOT NULL,
 salariu int
);

insert into manager(nume, salariu) values ('Mesia', 3000);
```

## Cerinta 14+15:

- ```
/*CERINTA 14 */  
  
/*Creaza o vizualizare cu toti inginerii care lucreaza pentru constructorul cu numarul 1*/  
  
create or replace view Vizualizare  
as  
select i.idinginer,i.nume,i.preume,i.salariu,i.data_angajare,i.data_expirare, i.idconstructor, e.nume as 'NumeConstructor', e.buget  
from inginer i join constructor e on (i.idconstructor = e.idconstructor)  
where i.idconstructor = 1;  
  
/* Urmatorul insert este permis deoarece se modifica doar tabelul cu cheia primara, adica inginer*/  
  
insert into Vizualizare (idinginer, nume, preume, salariu, data_angajare, data_expirare, idconstructor) values (8, 'Pipsick', 'Daniel', 8000, '2018-04-20', '2021-05-21', 1);  
  
/* Urmatorul insert nu este permis deoarece se incearca modificare ambelor tabele, si constructor si inginer*/  
  
insert into Vizualizare (idinginer, nume, preume, salariu, data_angajare, data_expirare, idconstructor, NumeConstructor) values (8, 'Pipsick', 'Daniel', 8000, '2018-04-20', '2021-05-21', 1, 'Renault');  
  
rollback;  
select * from Vizualizare;  
  
/* CERINTA 15 */  
  
create index full_name  
on inginer (nume, preume);  
  
show index from inginer;  
  
select nume, preume, salariu  
from inginer  
where nume = 'Constantin';  
  
explain select nume, preume, salariu  
from inginer  
where nume = 'Constantin';
```

Cerinta 16:

```
/* CERINTA 16 */
```

```
/* Sa se arate respectivele selecturi, pentru toate circuitele posibile, chiar daca nu s-a efectuat o cursa pe acestea*/
```

- ```
select c.idcircuit,p.idsofer,p.pozitiefinala, p.calificare,p.tipcauciuc,e.numardrs, d.numarcurse, e.idlocatie,e.lungime,d.data_incepere
from cursa c left outer join campionat d on(c.ancampionat = d.ancampionat)
join circuit e on (c.idcircuit = e.idcircuit)
left outer join cursa_sofer p on (p.idcircuit = c.idcircuit);
```

```
/*DIVISION*/
```

```
/*1. Sa se afiseze inginerii care au lucrat la toate motoarele care au mai putin de 820 de cai putere, cu count*/
```

- ```
select idinginer
from motor_inginer
where codmotor in(
    Select codmotor from motor
    where caiputere<820
)
group by idinginer
having count(idinginer) = (select count(*) from motor where caiputere<820)
;
```

```
/* 2.Acelasi select de mai sus, cu not exists*/
```

- ```
select distinct idinginer
from motor_inginer m
where not exists(
 select *
 from motor s
 where caiputere<820 and not exists (
 select 'x'
 from motor_inginer m2
 where s.codmotor = m2.codmotor and m.idinginer = m2.idinginer
)
);
```