1.

a)

```
CREATE TABLE Persoana (
   id_p INT PRIMARY KEY,
   nume VARCHAR(25),
   email VARCHAR(25),
   adresa VARCHAR(25)
);
```

b)

```
CREATE TABLE Contract_j (
   id_cj  NUMBER(6,0) PRIMARY KEY,
   data DATE,
   obiect VARCHAR(25),
   onorar DECIMAL(10, 2),
   nr_pagini NUMBER(6,0),
   id_client NUMBER(6,0),
   id_avocat NUMBER(6,0),
   FOREIGN KEY (id_client) REFERENCES Persoana(id_p),
   FOREIGN KEY (id_avocat) REFERENCES Persoana(id_p)
);
```

c)

```
CREATE TABLE Contract_m (
   id_cm NUMBER(6,0) PRIMARY KEY,
   data DATE,
   functie VARCHAR(25),
   salar_baza DECIMAL(10, 2),
   comision DECIMAL(5, 2),
   id_angajat NUMBER(6,0),
   FOREIGN KEY (id_angajat) REFERENCES Persoana(id_p)
);
```

d)

```
CREATE TABLE Rata (
   id_cj NUMBER(6,0),
   id_r NUMBER(6,0),
   data DATE,
```

```
  suma DECIMAL(10, 2),
  PRIMARY KEY (id_cj, id_r),
  FOREIGN KEY (id_cj) REFERENCES Contract_j(id_cj)
);
```

e) deja adaugat

f)

```
ALTER TABLE Contract_j
ADD tva DECIMAL(5, 2) AS (onorar * 0.19);
```

2. Să se exprime următoarele constrângeri (la nivel atribut sau tuplă):

a) Atributul onorar trebuie să ia valori între 100 și 20000.

```
ALTER TABLE Contract_j
ADD CONSTRAINT onorar_range_chk CHECK (onorar BETWEEN 100 AND 20000);
```

Example:

```
INSERT INTO Contract_j (id_cj, data, obiect, onorar, nr_pagini, id_client, id_avocat)
VALUES (2, TO_DATE('2023-05-15', 'YYYY-MM-DD'), 'Contract Sale',  10, 1, 2, 20);
```

b) Dacă funcția unui angajat este diferită de ‚avocat' atunci comisionul trebuie să fie NULL.

```
ALTER TABLE Contract_m
ADD CONSTRAINT functie_comision_chk CHECK ((functie != 'avocat' AND comision IS NULL) OR functie = 'avocat');
```

Example :

~~~~~~~~~~~ NU MERGE ~~~~~~~~~~~

```
INSERT INTO Contract_m (id_cm, data, functie, salar_baza, comision, id_angajat) VALUES (1, TO_DATE('2023-05-15', 'YYYY-MM-DD'), 'manager', 5000, 1, 1);
```

~~~~~~~~~~~ MERGE ~~~~~~~~~~~

```
INSERT INTO Contract_m (id_cm, data, functie, salar_baza, comision, id_angajat) VALUES (1, TO_DATE('2023-05-15', 'YYYY-MM-DD'), 'manager', 5000, NULL, 1);
```

3. Să se exprime în SQL următoarele interogări:

a) Să se găsească detaliile pentru contractele de asistență juridică din anul 2022 ce au numărul de pagini cuprins între 10 și 20.

```
SELECT *
FROM Contract_j
WHERE data BETWEEN TO_DATE('2022-01-01', 'YYYY-MM-DD') AND TO_DATE('2022-12-31', 'YYYY-MM-DD') AND nr_pagini > 9 AND nr_pagini < 21;
```

b) Să se găsească detaliile contractelor de muncă în ordine descrescătoare a salariului de bază pentru contractele de muncă cu comision 40%.

```
SELECT *
FROM Contract_m
WHERE comision = 40
ORDER BY salar_baza DESC;
```

4. Să se exprime în SQL următoarele interogări folosind operatorul JOIN:

a) Să se găsească numele clienților pentru avocatul 'Ionescu George'.

```
SELECT p.nume AS nume_client, a.nume AS nume_avocat
FROM Persoana p
JOIN Contract_j cj ON p.id_p = cj.id_client
JOIN Persoana a ON cj.id_avocat = a.id_p
WHERE a.nume = 'Ionescu George';
```

b) Să se găsească (id_cj, id_r1, data1, suma1, id_r2, data2, suma2) pentru fiecare id_cj ce are cel puțin o rată achitată, cu condiția id_r1 și id_r2 sunt consecutive (1 cu 2 sau 3 cu 4.. etc.). Pentru număr impar de rate id_r2, data2, suma2 sunt necompletate.

```
SELECT DISTINCT cj.id_cj, r1.id_r AS id_r1, r1.data AS data1, r1.suma AS suma1,
    NVL(r2.id_cj, '') AS id_cj2,NVL(r2.id_r, '') AS id_r2, NVL(r2.data, '') AS data2, NVL(r2.suma, '') AS suma2
FROM Contract_j cj
JOIN Rata r1 ON cj.id_cj = r1.id_cj
LEFT JOIN Rata r2 ON r1.id_cj = r2.id_cj AND r1.id_r + 1 = r2.id_r
```

5. Să se exprime în SQL fără funcții de agregare următoarele interogări folosind cel puţin o interogare imbricată şi operatori de genul EXISTS, IN, ALL, ANY:

a) Să se găsească contractele de asistență juridică cu onorar mai mare decât onorarul contractului juridic cu id_cj = 123.

```
SELECT *
FROM Contract_j
WHERE obiect = 'asistență juridică'
AND onorar > ALL (
   SELECT onorar
   FROM Contract_j
   WHERE id_cj = 123
);
```

b) Să se găsească numele avocaților care au exact un contract de asistență juridică.

```
SELECT p.nume AS nume_avocat,id_p AS id
FROM Persoana p
WHERE EXISTS (
   SELECT 1
   FROM Contract_j cj1
   WHERE cj1.id_avocat = p.id_p
   AND cj1.obiect = 'asistență juridică'
   GROUP BY cj1.id_avocat
   HAVING COUNT(id_cj) = 1
);
```

6. Să se exprime în SQL următoarele interogări folosind funcţii de agregare:

a) Să se găsească pentru fiecare nume de avocat valoarea medie a salariului pe

anul 2022.

```
SELECT p.nume AS nume_avocat,
    cm.id_cm AS id_contract_munca,
    cm.data AS data_contract_munca,
    cm.salar_baza AS salariu_baza,
    ROUND(SUM(cj.onorar * cm.comision / 100), 2) AS suma_comisioane,
    ROUND((cm.salar_baza - SUM(cj.onorar * cm.comision / 100)), 2) AS salariu_total
FROM Persoana p
JOIN Contract_j cj ON p.id_p = cj.id_avocat
JOIN Contract_m cm ON cj.id_avocat = cm.id_angajat
WHERE cj.data BETWEEN TO_DATE('2022-01-01', 'YYYY-MM-DD') AND TO_DATE('2022-12-31', 'YYYY-MM-DD')
GROUP BY p.nume, cm.id_cm, cm.data, cm.salar_baza;
```

b) Să se găsească id_cj pentru contractele juridice neachitate în întregime (suma

ratelor nu acoperă onorariul).

```
SELECT cj.id_cj, cj.data, cj.obiect, cj.onorar, ROUND(SUM(r.suma), 2) AS suma_achitata
FROM Contract_j cj
LEFT JOIN Rata r ON cj.id_cj = r.id_cj
GROUP BY cj.id_cj, cj.data, cj.obiect, cj.onorar
HAVING ROUND(SUM(r.suma), 2) < cj.onorar;
```

7. Să se scrie instrucţiunile pentru actualizarea BD:

a) Să se adauge contractul de muncă cu numărul 80 din data ’01-SEP-2022’,npentru avocatul ‘Andrei Vasile’ cu număr de telefon ‘0545-123456’, identificator de angajat 13, salariu de bază 10000 și comision 12,5.

```
INSERT INTO Persoana (id_p, nume, email, adresa)
VALUES (13, 'Andrei Vasile', '0545-123456' , '31 SF');

INSERT INTO Contract_m (id_cm, data, functie, salar_baza, comision, id_angajat)
VALUES (80, TO_DATE('01-SEP-2022', 'DD-MON-YYYY'), 'avocat', 10000, 12.5, 13);
```

b) Să se șteargă contractele juridice mai vechi de 1 an față de ’01-SEP-2022’ ce

nu au rate achitate.

```
INSERT INTO Contract_j (ID_CJ, DATA, OBIECT, ONORAR, NR_PAGINI, ID_CLIENT, ID_AVOCAT)

VALUES (124, TO_DATE('01-AUG-2021', 'DD-MON-YYYY'), 'asistență juridică', 2000, 15, 2, 13);
```

~~~~~~~~~Check insert

```
SELECT *

FROM Contract_j
```

~~~~~~~~~

```
DELETE FROM Contract_j

WHERE data < TO_DATE('01-SEP-2021', 'DD-MON-YYYY')

AND id_cj NOT IN (

  SELECT id_cj

  FROM Rata

);
```

~~~~~Verificare

```
SELECT *

FROM Contract_j
```

c) Să se modifice salariul de bază al avocaților, crescând cu 2%, pentru avocații

cu vechime în muncă peste 5 ani.

~~~CHECK

```
SELECT *

FROM Contract_m

Where id_cm = 3;
```

~~UPDATE

```
UPDATE Contract_m

SET salar_baza = salar_baza * 1.02

WHERE functie = 'avocat'

AND (MONTHS_BETWEEN(CURRENT_DATE, data) / 12) > 5;
```

~~~~VERIFY

```
SELECT *

FROM Contract_m

Where id_cm = 3;
```

9.Să se definească triggere pentru:

a) A asigura că la modificare salar de bază angajat, valoarea nouă este mai mare

decât valoarea precedentă.

~~~TRIGGER

```
CREATE OR REPLACE TRIGGER CheckSalarBaza
BEFORE UPDATE OF salar_baza ON Contract_m
FOR EACH ROW
BEGIN
   IF :NEW.salar_baza <= :OLD.salar_baza THEN
      RAISE_APPLICATION_ERROR(-20001, 'new salar_baza is greater.');
   END IF;
END;
```

~~~~Valoarea salariului lui id_cm = 5

```
SELECT id_cm,salar_baza
FROM Contract_m
Where id_cm=5;
```

~~~~Verify Error

```
UPDATE Contract_m
SET salar_baza = 1200.0
WHERE id_cm = 5;
```
~~~~