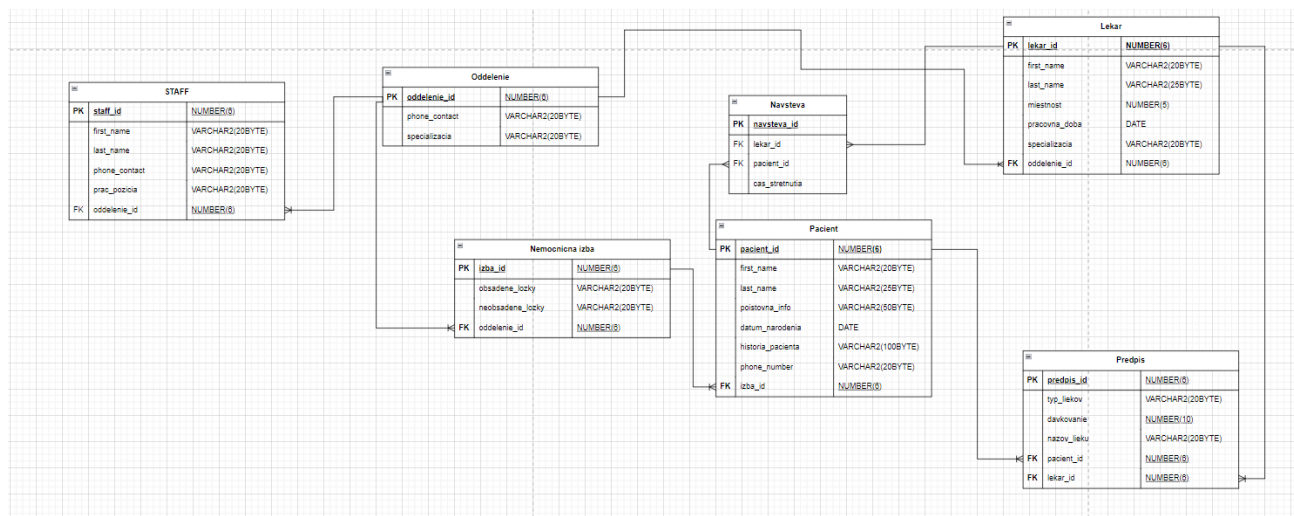


Úvod

Venovala som DB pre nemocnicu. Takáto databáza môže zahŕňať rôzne aspekty nemocničných operácií, ako sú údaje o pacientoch, lekároch a navstevách, vypísaných predpisoch; údaje o oddeleniach a nemocničných izbách, ktoré nachádzajú v jednotlivých oddeleniach, takže informácie o STAFF (ľudia, ktorí pracujú v nemocnici, okrem lekárov)

Zameriavala som sa na vytvorenie rôznych pohľadov a triggerov. Nižšie je zhrnutie každej úlohy spolu s SQL kódom a demonštráciou funkčnosti.



Druhá časť zadania

- 2 pohľady s netriviálnym selektom nad jednou tabuľkou (nestačí použiť iba vymenovanie stĺpcov, treba použiť "niečo navyše" napr.: vstavané funkcie)

```
-- vypočíta vek pacientov
CREATE VIEW Pacient_Vek AS
SELECT
    pacient_id,
    first_name,
    last_name,
    FLOOR((SYSDATE - datum_narodenia) / 365) AS vek,
    izba_id
FROM
    Pacient;
```

PACIENT_ID	FIRST_NAME	LAST_NAME	VEK	IZBA_ID
1	Ivan	Petrov	33	1
2	Maria	Sidorova	38	2
3	Daria	Vershynina	81	3
4	Roman	Hulyk	80	5
5	Nastia	Radishevska	28	4
6	Anna	Leonenko	52	6
7	Polina	Rapp	32	7
8	Nikita	Demianenko	59	8
9	Dushan	Herich	17	9
10	Natalia	Portman	20	10
11	Ivan	Petrov	5	1

```
-- počet pacientov v izbách nemocnici
CREATE VIEW Pocet_Pacientov_Izba AS
SELECT
    izba_id,
    COUNT(pacient_id) AS pocet_pacientov
FROM
    Pacient
GROUP BY
    izba_id;
```

IZBA_ID	POCET_PACIENTOV
6	1
1	2
7	1
2	1
8	1
5	1
4	1
10	1
3	1
9	1

- 3 pohľady so spájaním tabuliek (1x spojenie 3 tabuliek, 1x outer join, 1x spojenie aspoň 2 tabuliek)

```
-- Pohľad so spájaním troch tabuliek(Pacient, Lekar a Navsteva)
CREATE VIEW NavstevaDetail AS
SELECT
    n.navsteva_id,
    n.cas_stretnutia,
    p.first_name AS pacient_first_name,
    p.last_name AS pacient_last_name,
    l.first_name AS lekar_first_name,
    l.last_name AS lekar_last_name
FROM
    Navsteva n
INNER JOIN
    Pacient p ON n.pacient_id = p.pacient_id
INNER JOIN
    Lekar l ON n.lekar_id = l.lekar_id;
```

NAVSTEVA_ID	CAS_STRETNUTIA	PACIENT_FIRST_NAME	PACIENT_LAST_NAME	LEKAR_FIRST_NAME	LEKAR_LAST_NAME
1	15-APR-24	Ivan	Petrov	Elena	Sidorova
2	16-APR-24	Maria	Sidorova	Pavel	Ivanov
3	17-APR-24	Daria	Vershynina	Michael	Johnson
4	18-APR-24	Roman	Hulyk	Anna	Kovacs
5	19-APR-24	Nastia	Radishevska	Sophia	Li
6	20-APR-24	Anna	Leonenko	Daniel	Garcia
7	21-APR-24	Polina	Rapp	Emma	Chen
8	22-APR-24	Nikita	Demianenko	Alex	Wong
9	23-APR-24	Dushan	Herich	Maria	Gonzalez
10	24-APR-24	Natalia	Portman	Valeria	Buhaiova

```
-- Pohľad s outer join( spája tabuľky Pacient a Predpis left join)
CREATE VIEW PacientPredpis AS
SELECT
    p.pacient_id,
    p.first_name,
    p.last_name,
    pr.predpis_id,
    pr.typ_liekov,
    pr.nazov_lieku
FROM
    Pacient p
LEFT OUTER JOIN
    Predpis pr ON p.pacient_id = pr.pacient_id;
```

PACIENT_ID	FIRST_NAME	LAST_NAME	PREDPIS_ID	TYP_LIEKOV	NAZOV_LIEKU
1	Ivan	Petrov	1	Tabletky	Aspirin
2	Maria	Sidorova	2	Injekcia	Nimesil
3	Daria	Vershynina	3	Sirup	Paracetamol
4	Roman	Hulyk	4	Tabletky	Ibuprofen
5	Nastia	Radishevska	5	Kapsule	Amoxicillin
6	Anna	Leonenko	6	Injekcia	Dexamethasone
7	Polina	Rapp	7	Tabletky	Lorazepam
8	Nikita	Demianenko	8	Sirup	Cetirizine
9	Dushan	Herich	9	Tabletky	Omeprazole
10	Natalia	Portman	10	Injekcia	Morphine

```
-- Pohľad so spojením dvoch tabuliek
CREATE VIEW LekarOddelenie AS
SELECT
    l.lekar_id,
    l.first_name,
    l.last_name,
    l.specializacia,
    o.specializacia AS oddelenie_specializacia
FROM
    Lekar l
INNER JOIN
    Oddelenie o ON l.oddelenie_id = o.oddelenie_id;
```

LEKAR_ID	FIRST_NAME	LAST_NAME	SPECIALIZACIA	ODDELENIE_SPECIALIZACIA
1	Elena	Sidorova	Kardiologia	Kardiologia
2	Pavel	Ivanov	Ortopedia	Ortopedia
3	Michael	Johnson	Neurology	Ortopedia
4	Anna	Kovacs	Pediatrics	Neurologia
5	Sophia	Li	Obstetrics	Gynekologia
6	Daniel	Garcia	Psychiatry	Oftalmologia
7	Emma	Chen	Dermatology	Urologia
8	Alex	Wong	Emergency Medicine	Dermatologia
9	Maria	Gonzalez	Urology	Psychiatria
10	Valeria	Buhaiova	Ortoped	Psychiatria

3. 2 pohľady s použitím agregáčnych funkcií alebo zoskupenia

```
-- Počet pacientov v jednotlivých oddeleniach(zoskupenie-oddelenia,agreguje-pacientov v oddeleniach)
CREATE VIEW Pacienti_v_oddeleniach AS
SELECT O.specializacia, COUNT(P.pacient_id) AS pocet_pacientov
FROM Pacient P
JOIN Nemocnicna_izba NI ON P.izba_id = NI.izba_id
JOIN Oddelenie O ON NI.oddelenie_id = O.oddelenie_id
GROUP BY O.specializacia;
```

SPECIALIZACIA	POCET_PACIENTOV
Urologia	1
Kardiologia	2
Pediatrica	1
Dermatologia	1
Ortopedia	1
Oftalmologia	1
Neurologia	1
Oncologia	1
Gynekologia	1
Psychiatria	1

```
-- Počet predpisov podľa lekárov(zoskupenie podľa lekára a agreguje počet predpisov)
CREATE VIEW Predpisy_podla_lekarov AS
SELECT L.first_name || ' ' || L.last_name AS meno_lekara, COUNT(Pr.predpis_id) AS pocet_predpisov
FROM Predpis Pr
JOIN Lekar L ON Pr.lekar_id = L.lekar_id
GROUP BY L.first_name, L.last_name;
```

MENO_LEKARA	POCET_PREDPISOV
Michael Johnson	1
Emma Chen	1
Alex Wong	1
Maria Gonzalez	1
Valeria Buhaiova	1
Daniel Garcia	1
Pavel Ivanov	1
Anna Kovacs	1
Elena Sidorova	1
Sophia Li	1

4. 1 pohľad s použitím množinových operácií

```
-- UNION-pacientov a lekárov, ktorí sú priradení k oddeleniu s ID = 1
CREATE VIEW Oddelenie_1_People AS
SELECT first_name, last_name, 'Pacient' AS type
FROM Pacient
WHERE izba_id IN (
    SELECT izba_id FROM Nemocnica_izba WHERE oddelenie_id = 1
)
UNION
SELECT first_name, last_name, 'Lekár' AS type
FROM Lekar
WHERE oddelenie_id = 1;
```

FIRST_NAME	LAST_NAME	TYPE
Elena	Sidorova	Lekár
Ivan	Petrov	Pacient

5. 2 pohľady s použitím netriviálnych vnorených selektov

```
-- poskytuje informácie o lekároch, navštevách, a počet pacientov, ktorí majú naplánované navštevy s lekármi
CREATE VIEW Lekari_Pacienti_Navstevy AS
SELECT
    l.lekar_id,
    l.first_name,
    l.last_name,
    l.specializacia,
    (SELECT COUNT(*)
     FROM Navsteva v
     WHERE v.lekar_id = l.lekar_id) AS pocet_navstev
FROM
    Lekar l;
```

LEKAR_ID	FIRST_NAME	LAST_NAME	SPECIALIZACIA	POCET_NAVSTEV
1	Elena	Sidorova	Kardiologia	1
2	Pavel	Ivanov	Ortopedia	1
3	Michael	Johnson	Neurology	1
4	Anna	Kovacs	Pediatrics	1
5	Sophia	Li	Obstetrics	1
6	Daniel	Garcia	Psychiatry	1
7	Emma	Chen	Dermatology	1
8	Alex	Wong	Emergency Medicine	1
9	Maria	Gonzalez	Urology	1
10	Valeria	Buhaiova	Ortoped	1

```
-- Údaje o zamestnancoch a ich oddeleniach(Pomocou vnoreného selektu sa získava celkový počet zamestnancov v každom oddelení)
CREATE VIEW Staff_Oddelenie AS
SELECT
    s.staff_id,
    s.first_name AS staff_first_name,
    s.last_name AS staff_last_name,
    s.prac_pozicia,
    o.specializacia,
    (SELECT COUNT(*) FROM Staff ss WHERE ss.oddelenie_id = o.oddelenie_id) AS pocet_zamestnancov
FROM
    Staff s
JOIN
    Oddelenie o ON s.oddelenie_id = o.oddelenie_id
WHERE
    s.prac_pozicia = 'Sestra';
```

STAFF_ID	STAFF_FIRST_NAME	STAFF_LAST_NAME	PRAC_POZICIA	SPECIALIZACIA	POCET_ZAMESTNANCOV
1	Maria	Ivanova	Sestra	Kardiologia	1
2	Petr	Sidorov	Sestra	Ortopedia	1
3	Olga	Petrova	Sestra	Oncologia	1
4	Ivan	Kuznetsov	Sestra	Neurologia	1
5	Elena	Popova	Sestra	Pediatrica	1
6	Andrei	Vasiliev	Sestra	Gynekologia	1
7	Tatiana	Smirnova	Sestra	Oftalmologia	1
8	Sergei	Mikhailov	Sestra	Urologia	1
9	Natalia	Novikova	Sestra	Dermatologia	1
10	Dmitri	Sokolov	Sestra	Psychiatria	1

6. vytvorte 1 sekvenciu na generovanie primárnych kľúčov a triggre, ktorá bude vkladať hodnoty do príslušných tabuliek

```
-- Sekvencia na generovanie jedinečných ID pre tabuľku Nemocnicna_izba
CREATE SEQUENCE nemocnicna_izba_seq
  START WITH 11
  INCREMENT BY 1;

-- Trigger na automatické priradenie ID zo sekvencie pred vložení záznamu
CREATE OR REPLACE TRIGGER trg_nemocnicna_izba
  BEFORE INSERT ON Nemocnicna_izba
  FOR EACH ROW
BEGIN
  IF :NEW.izba_id IS NULL THEN
    :NEW.izba_id := nemocnicna_izba_seq.NEXTVAL;
  END IF;
END;
/

-- test vlozenim novych hodnt bez primary key
INSERT INTO Nemocnicna_izba (obsadene_lozky, neobsadene_lozky, oddelenie_id)
VALUES ('5', '15', 1);
Select*from Nemocnicna_izba;
```

Trigger created.

1 row(s) inserted.

IZBA_ID	OBSADENE_LOZKY	NEOBSADENE_LOZKY	ODDELENIE_ID
1	5	15	1
2	8	12	2
3	9	11	3
4	1	19	4
5	7	13	5
6	4	16	6
7	3	17	7
8	10	10	8
9	2	18	9
10	11	9	10
11	5	15	1

7. vytvorte 1 ľubovoľný trigger okrem typu triggera uvedeného v predchádzajúcom bode (trigger musí obsahovať buď podmienku, cyklus alebo výnimku)

```
--trigger s podmienkou
CREATE OR REPLACE TRIGGER ensure_unique_names
BEFORE INSERT OR UPDATE
ON STAFF
FOR EACH ROW
BEGIN
    IF :NEW.first_name = :NEW.last_name THEN
        RAISE_APPLICATION_ERROR(-20001, 'First name and last name cannot be the same.');
```

```
    END IF;
END;
/

INSERT INTO STAFF (staff_id, first_name, last_name, phone_contact, prac_pozicia, oddelenie_id)
VALUES (11, 'Maria', 'Maria', '111-222', 'Sestra', 1);
Select*from STAFF ;
```

Trigger created.

ORA-20001: First name and last name cannot be the same. ORA-06512: at "SQL_BUTTGJLXJZSXDIRYDDWXSXMT.ENSURE_UNIQUE_NAMES", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

STAFF_ID	FIRST_NAME	LAST_NAME	PHONE_CONTACT	PRAC_POZICIA	ODDELENIE_ID
1	Maria	Ivanova	111-222	Sestra	1
2	Petr	Sidorov	333-444	Sestra	2
3	Olga	Petrova	555-666	Sestra	3
4	Ivan	Kuznetsov	777-888	Sestra	4
5	Elena	Popova	999-000	Sestra	5
6	Andrei	Vasiliev	111-222	Sestra	6
7	Tatiana	Smirnova	333-444	Sestra	7
8	Sergei	Mikhailov	555-666	Sestra	8
9	Natalia	Novikova	777-888	Sestra	9
10	Dmitri	Sokolov	999-000	Sestra	10

Záver

Netriviálne selekty	1
Spájanie tabuliek	3
Použitie agregáčnych funkcií alebo zoskupenia	2
Množinová operácia	1
Vnorené selekty	2
Sekvencia + trigger	1
Trigger	1
Celkovo	11