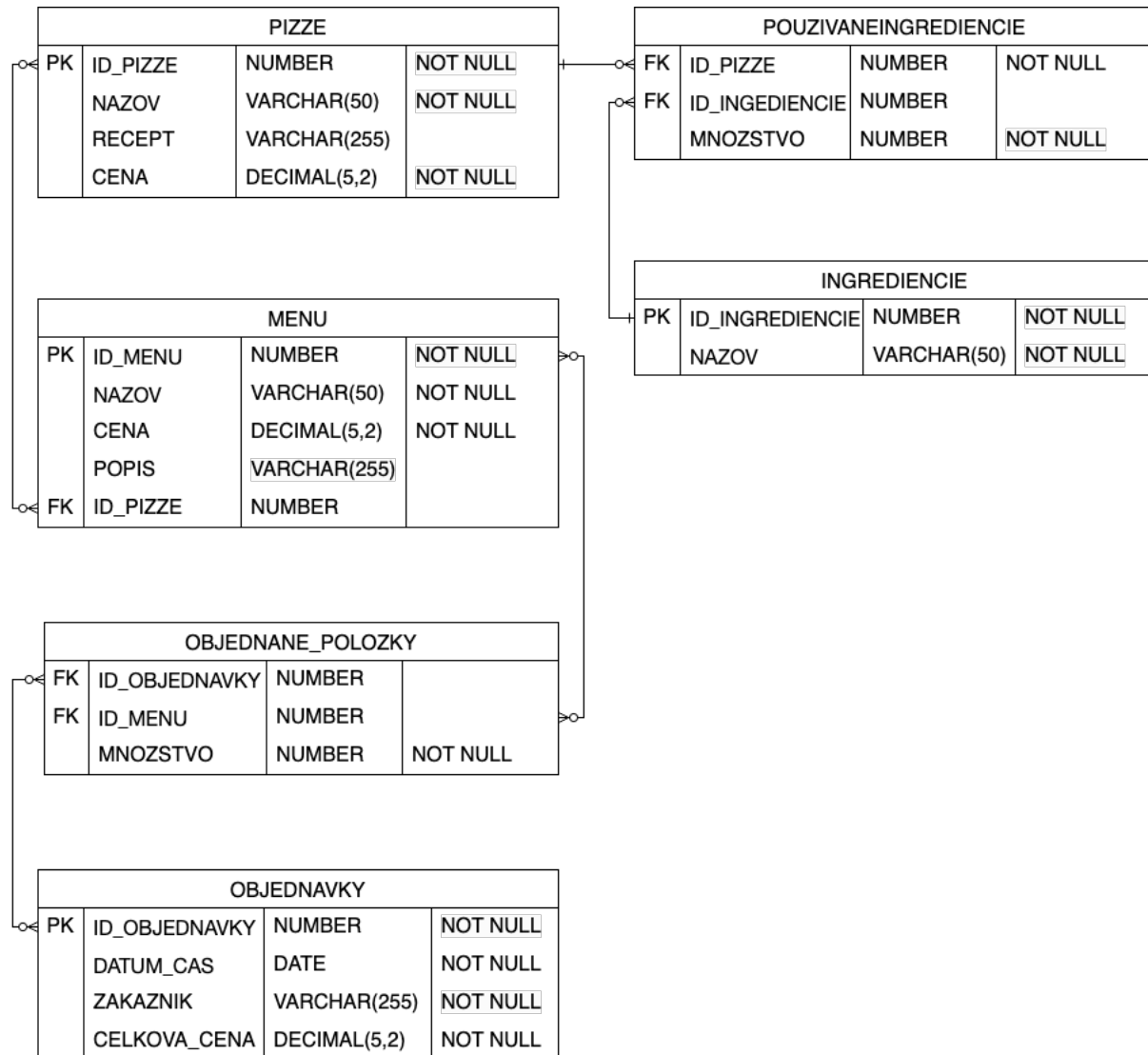


Úvod

Hlavnou úlohou tohto zadania bolo vytvoriť databázu menu pre pizzeriu, v ktorej by bolo možné spravovať celý sortiment podniku, potrebné ingrediencie pre všetky druhy pizze, vytvárať objednávky a spravovať ich jednotlivé položky.



Druhá časť zadania

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

-- Vypíšeme názov a cenu (k cene pridáme znak EURO) všetkých pizze:

```
CREATE VIEW CENA_PIZZE AS  
SELECT NAZOV, CONCAT('€', cena) AS CENA  
FROM PIZZE;
```

```
SELECT * FROM CENA_PIZZE;
```

SQL Worksheet

```
103  
104 ✓ CREATE VIEW CENA_PIZZE AS  
105 SELECT NAZOV, CONCAT('€', cena) AS CENA  
106 FROM PIZZE;  
107  
108 SELECT * FROM CENA_PIZZE;  
109
```

View created.

NAZOV	CENA
Margherita	€5.5
Pepperoni	€6.5
Capricciosa	€7.5
Vegetariánska	€7.5
Hawai	€6.5

-- Vypíšeme počet objednávek za každý deň:

```
CREATE VIEW POCET_OBJ AS  
SELECT DATUM_CAS AS DATUM, COUNT(*) AS POCET  
FROM OBJEDNAVKY  
GROUP BY DATUM_CAS  
ORDER BY DATUM;
```

```
SELECT * FROM POCET_OBJ;
```

SQL Worksheet

```
104 ✓ CREATE VIEW POCET_OBJ AS  
105 SELECT DATUM_CAS AS DATUM, COUNT(*) AS POCET  
106 FROM OBJEDNAVKY  
107 GROUP BY DATUM_CAS  
108 ORDER BY DATUM;  
109  
110 SELECT * FROM POCET_OBJ;
```

View created.

DATUM	POCET
20-MAR-23	1
21-MAR-23	1
22-MAR-23	1
23-MAR-23	1
24-MAR-23	1

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

-- Vypíšeme potrebné množstvo ingrediencií na každú pizzu: (*spojenie 3 tabuliek*)

```
SELECT P.NAZOV, I.NAZOV, PI.MNOZSTVO FROM PIZZE P
JOIN POUZIVANEINGREDIENCIE PI ON P.ID_PIZZE = PI.ID_PIZZE
JOIN INGREDIENCIE I ON PI.ID_INGREDIENCIE = I.ID_INGREDIENCIE
ORDER BY P.NAZOV;
```

SQL Worksheet

```
104 v CREATE VIEW PIZZE_INGR AS
105 SELECT P.NAZOV, I.NAZOV, PI.MNOZSTVO FROM PIZZE P
106 JOIN POUZIVANEINGREDIENCIE PI ON P.ID_PIZZE = PI.ID_PIZZE
107 JOIN INGREDIENCIE I ON PI.ID_INGREDIENCIE = I.ID_INGREDIENCIE
108 ORDER BY P.NAZOV ;
109 |
110 SELECT * FROM PIZZE_INGR;
```

NAZOV	NAZOV	MNOZSTVO
Capricciosa	Syr mozzarella	2
Capricciosa	Šunka	2
Capricciosa	Paradajkový základ	1
Capricciosa	Šampiňóny	2
Hawai	Syr mozzarella	2
Hawai	Šampiňóny	1

-- Vypíšeme prvé tri údaje objednávky, ktoré budú obsahovať ID objednávky, dátum a vybrané menu: (*outer join*)

```
CREATE VIEW OBJ_DETAIL AS
SELECT O.ID_OBJEDNAVKY, O.DATUM_CAS, M.NAZOV AS MENU_NAZOV
FROM OBJEDNAVKY O
LEFT OUTER JOIN OBJEDNANEPOLOZKY OP ON O.ID_OBJEDNAVKY = OP.ID_OBJEDNAVKY
LEFT OUTER JOIN MENU M ON OP.ID_MENU = M.ID_MENU
LEFT OUTER JOIN PIZZE P ON M.ID_PIZZE = P.ID_PIZZE
FETCH FIRST 3 ROWS ONLY;

SELECT * FROM OBJ_DETAIL;
```

SQL Worksheet

```
104 ✓ CREATE VIEW OBJ_DETAIL AS
105 SELECT O.ID_OBJEDNAVKY, O.DATUM_CAS, M.NAZOV AS MENU_NAZOV
106 FROM OBJEDNAVKY O
107 LEFT OUTER JOIN OBJEDNANEPOLOZKY OP ON O.ID_OBJEDNAVKY = OP.ID_OBJEDNAVKY
108 LEFT OUTER JOIN MENU M ON OP.ID_MENU = M.ID_MENU
109 LEFT OUTER JOIN PIZZE P ON M.ID_PIZZE = P.ID_PIZZE
110 FETCH FIRST 3 ROWS ONLY;
111
112
113 SELECT * FROM OBJ_DETAIL;
```

ID_OBJEDNAVKY	DATUM_CAS	MENU_NAZOV
1	20-MAR-23	Pizza Pepperoni + cola
2	21-MAR-23	Pizza Pepperoni + cola
3	22-MAR-23	Pizza Vegetariánska + cola

Download CSV

-- Vypíšeme množstvo ingrediencií použitých vo všetkých pizzách: (*spojenie aspoň 2 tabuliek*)

```
CREATE VIEW POCET_POUZITICH_ING AS
SELECT I.NAZOV AS INGREDIENCIA_NAZOV, COUNT(PI.ID_INGREDIENCIE) AS
POCET_POUZITI
FROM INGREDIENCIE I
JOIN POUZIVANEINGREDIENCIE PI ON I.ID_INGREDIENCIE = PI.ID_INGREDIENCIE
GROUP BY I.NAZOV
ORDER BY POCET_POUZITI DESC;
```

```
SELECT * FROM POCET_POUZITICH_ING;
```

SQL Worksheet

```
104 ✓ CREATE VIEW POCET_POUZITICH_ING AS
105 SELECT I.NAZOV AS INGREDIENCIA_NAZOV, COUNT(PI.ID_INGREDIENCIE) AS POCET_POUZITI
106 FROM INGREDIENCIE I
107 JOIN POUZIVANEINGREDIENCIE PI ON I.ID_INGREDIENCIE = PI.ID_INGREDIENCIE
108 GROUP BY I.NAZOV
109 ORDER BY POCET_POUZITI DESC;
110
111
112
113 SELECT * FROM POCET_POUZITICH_ING;
```

INGREDIENCIA_NAZOV	POCET_POUZITI
Syr mozzarella	5
Paradajkový základ	5
Šampiňóny	3
Šunka	2
Saláma pepperoni	1

3. 2 pohľady s použitím agregáčnych funkcií alebo zoskupenia;

-- Vypíšeme priemerné ceny objednávky za každý deň:

```
CREATE VIEW PREMIERNA_CENA_PIZZ AS  
SELECT DATUM_CAS, AVG(CELKOVA_CENA) AS PREMIERNA_CENA  
FROM OBJEDNAVKY  
GROUP BY DATUM_CAS  
ORDER BY DATUM_CAS;
```

```
SELECT * FROM PREMIERNA_CENA_PIZZ;
```

SQL Worksheet

```
104 ✓ CREATE VIEW PREMIERNA_CENA_PIZZ AS  
105 SELECT DATUM_CAS, AVG(CELKOVA_CENA) AS PREMIERNA_CENA  
106 FROM Objednavky  
107 GROUP BY DATUM_CAS  
108 ORDER BY DATUM_CAS;  
109  
110  
111 SELECT * FROM PREMIERNA_CENA_PIZZ;
```

View created.

DATUM_CAS	PREMIERNA_CENA
20-MAR-23	10
21-MAR-23	10
22-MAR-23	10
23-MAR-23	10
24-MAR-23	10

-- Vypíšeme počet všech pizz v menu:

```
CREATE VIEW POCET_PIZZE AS  
SELECT COUNT(*) AS POCET_PIZZ FROM PIZZE;
```

```
SELECT * FROM POCET_PIZZE;
```

SQL Worksheet

```
103  
104 ✓ CREATE VIEW POCET_PIZZE AS  
105 SELECT COUNT(*) AS POCET_PIZZ FROM PIZZE;  
106  
107  
108 SELECT * FROM POCET_PIZZE;|
```

View created.

POCET_PIZZ
5

Download CSV

4. 1 pohľad s použitím množinových operácií

-- Vypíšeme názov pizze, ktorá končí na a, jej recept a cenu, ktorá je vyššia ako 6:

```
CREATE VIEW PIZZA_DETAIL AS
SELECT * FROM PIZZE WHERE CENA > 6
INTERSECT
SELECT * FROM PIZZE WHERE NAZOV LIKE ('%a');

SELECT * FROM PIZZA_DETAIL;
```

SQL Worksheet



```
104 CREATE VIEW PIZZA_DETAIL AS
105 SELECT * FROM PIZZE WHERE CENA > 6
106 INTERSECT
107 SELECT * FROM PIZZE WHERE NAZOV LIKE ('%a');
108
109 SELECT * FROM PIZZA_DETAIL;
```

View created.

ID_PIZZE	NAZOV	RECEPT	CENA
3	Capricciosa	Paradajkový základ, syr mozzarella, šunka, šampiňóny a olivy	7.5
4	Vegetariánska	Paradajkový základ, syr mozzarella, zelenina a olivy	7.5

Download CSV

2 rows selected.

5. 2 pohľady s použitím vnorených selektov

-- Vypíšeme názov, popis a cenu najlacnejšej pizze:

```
CREATE VIEW MIN_PIZZE AS  
SELECT NAZOV, POPIS, CENA FROM Menu  
WHERE CENA = (SELECT MIN(CENA) FROM MENU);  
  
SELECT * FROM MIN_PIZZE;
```

SQL Worksheet

```
102  
103 ✓ CREATE VIEW MIN_PIZZE AS  
104 SELECT NAZOV, POPIS, CENA FROM Menu  
105 WHERE CENA = (SELECT MIN(CENA) FROM MENU);  
106  
107 SELECT * FROM MIN_PIZZE;
```

View created.

NAZOV	POPIS	CENA
Pizza Margherita + cola	Classicka pizza	6

Download CSV

-- Vypíšeme id, názov, recept a cenu pizze, ktorá stojí viac ako priemerná cena pizze:

```
CREATE VIEW MORE_AVG_PRICE AS
SELECT * FROM PIZZE
WHERE CENA > (SELECT AVG(CENA) FROM PIZZE);

SELECT * FROM MORE_AVG_PRICE;
```

SQL Worksheet

```
102
103 ✓ CREATE VIEW MORE_AVG_PRICE AS
104 SELECT * FROM PIZZE
105 WHERE CENA > (SELECT AVG(CENA) FROM PIZZE);
106
107 SELECT * FROM MORE_AVG_PRICE;
```

View created.

ID_PIZZE	NAZOV	RECEPT	CENA
3	Capricciosa	Paradajkový základ, syr mozzarella, šunka, šampiňóny a olivy	7.5
4	Vegetariánska	Paradajkový základ, syr mozzarella, zelenina a olivy	7.5

Download CSV

2 rows selected.

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

-- sekvencia na vytvorenie id v tabuľke pizza:

```
CREATE SEQUENCE ID_PIZZE_SEQ
  START WITH 6
  INCREMENT BY 1
  NOCYCLE
  NOCACHE;
```

-- trigger na vloženie nových id do tabuľky Pizza spolu s novými údajmi:

```
CREATE OR REPLACE TRIGGER ID_PIZZE_TRIGGER
  BEFORE INSERT ON PIZZE
  FOR EACH ROW
  BEGIN
    SELECT ID_PIZZE_SEQ.NEXTVAL INTO :NEW.ID_PIZZE FROM DUAL;
  END;
/
```

-- test vložením nových hodnôt bez primary key:

```
INSERT INTO PIZZE (NAZOV, RECEPT, CENA) VALUES ('DIABLO', 'PARADAJKOVÝ ZÁKLAD,
SYR MOZZARELLA, ČILI, JALAPENOS, SALAMI', 5.5);
```

SQL Worksheet Clear Find Actions

```

103 CREATE SEQUENCE ID_PIZZE_SEQ
104     START WITH 6
105     INCREMENT BY 1
106     NOCYCLE
107     NOCACHE;
108
109 CREATE OR REPLACE TRIGGER ID_PIZZE_TRIGGER
110     BEFORE INSERT ON PIZZE
111     FOR EACH ROW
112     BEGIN
113         SELECT ID_PIZZE_SEQ.NEXTVAL INTO :NEW.ID_PIZZE FROM DUAL;
114     END;
115 /
116 INSERT INTO PIZZE (NAZOV, RECEPT, CENA) VALUES ('DIABLO', 'PARADAJKOVÝ ZÁKLAD, SYR MOZZARELLA, ČILI, JALAPENOS, SALAMI', 5.5);
117 |
118 SELECT * FROM PIZZE;
```

5	Hawai	Paradajkový základ, syr mozzarella, šunka a ananás	6.5
6	DIABLO	PARADAJKOVÝ ZÁKLAD, SYR MOZZARELLA, ČILI, JALAPENOS, SALAMI	5.5

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

-- trigger, ktorý kontroluje, aby pri vkladaní nových údajov do tabuľky nebol dátum starší ako dnešný dátum:

```
CREATE TRIGGER OBJEDNAVKY_DO_BUDUCNOSTY
BEFORE INSERT ON OBJEDNAVKY
FOR EACH ROW
DECLARE
    CURR_DATE DATE := SYSDATE;
BEGIN
    IF :NEW.DATUM_CAS > CURR_DATE THEN
        RAISE_APPLICATION_ERROR(-20003, 'NIE JE MOŽNÉ ZADAŤ OBJEDNÁVKU V BUDÚCNOSTI.');
```

```
    END IF;
```

```
END;
```

```
/
```

-- test pridávania budúcich dátumov:

```
INSERT INTO OBJEDNAVKY VALUES (6, DATE'2023-09-24', 'DELIVERY', 10);
```

SQL Worksheet

Clear Find

```
110
111 CREATE TRIGGER OBJEDNAVKY_DO_BUDUCNOSTY
112 BEFORE INSERT ON OBJEDNAVKY
113 FOR EACH ROW
114 DECLARE
115     CURR_DATE DATE := SYSDATE;
116 BEGIN
117     IF :NEW.DATUM_CAS > CURR_DATE THEN
118         RAISE_APPLICATION_ERROR(-20003, 'NIE JE MOŽNÉ ZADAŤ OBJEDNÁVKU V BUDÚCNOSTI.');
```

```
    END IF;
```

```
END;
```

```
/
```

```
122 INSERT INTO OBJEDNAVKY VALUES (6, DATE'2023-09-24', 'DELIVERY', 10);
123
```

Trigger created.

ORA-20003: NIE JE MOŽNÉ ZADAŤ OBJEDNÁVKU V BUDÚCNOSTI ORA-06512: at "SQL_BBMYPNPKJBINTJLOUUEMLTHU.OBJEDNAVKY_DO_BUDUCNOSTY", line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721

Záver

Netriviálne selekty	2/2
Spájanie tabuliek	3/3
Použitie agregáčnych funkcií alebo zoskupenia	2/2
Množinová operácia	2/2
Vnorené selekty	2/2
Sekvencia + trigger	2/2
Trigger	2/2
Celkovo	15/15