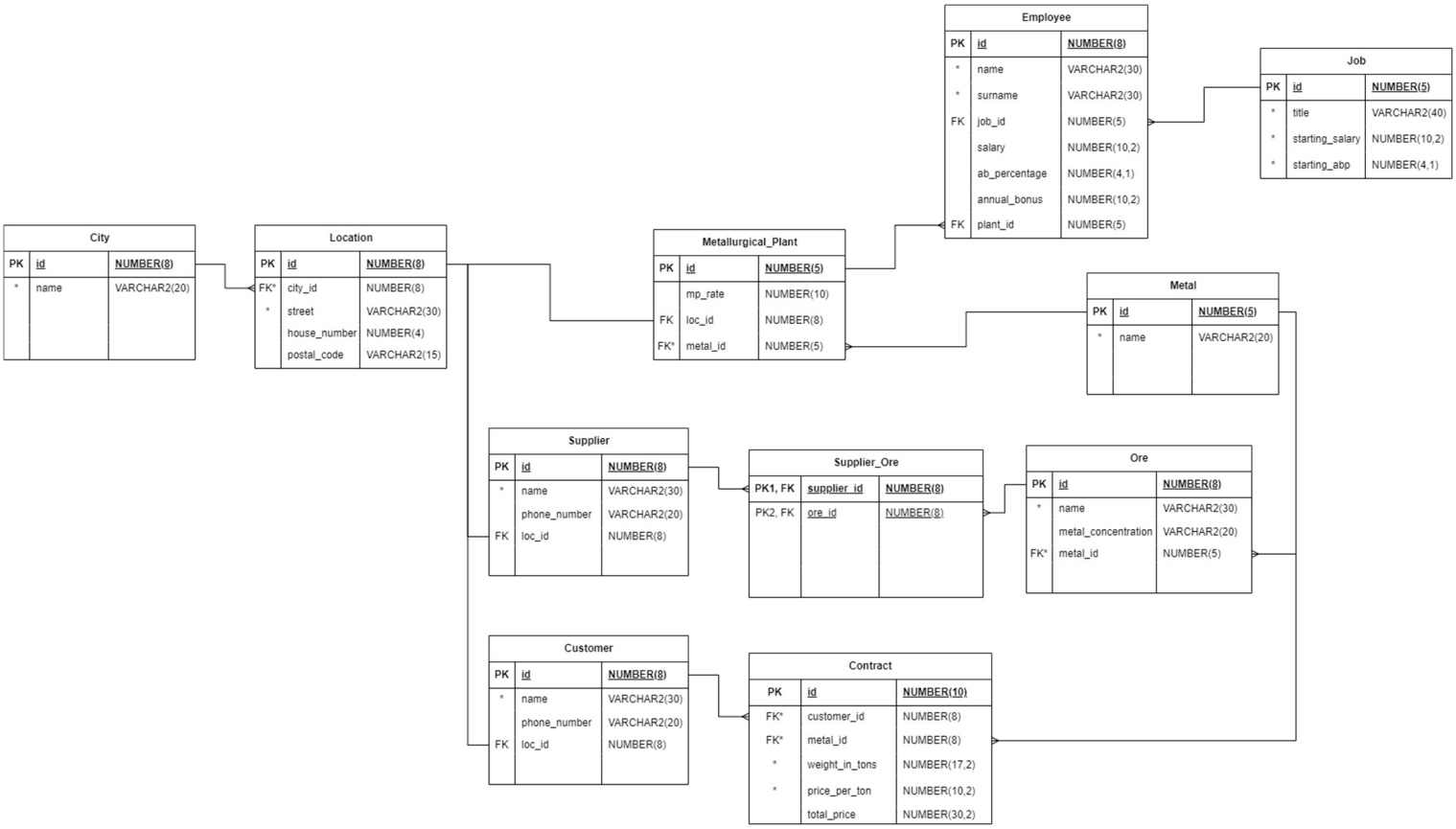
Úvod

Hlavným cieľom tejto úlohy bolo vytvoriť databázu pre hutnícky podnik. Databáza obsahuje tabuľky miest, lokalít, hutníckych závodov, dodávateľov, zákazníkov, zamestnancov, prac, rúd, kovov a zmlúv.



1. Vytváranie tabuliek z prvej časti zadania

```
CREATE TABLE City(
   id number(8) PRIMARY KEY,
   name varchar2(20) NOT NULL
);

CREATE TABLE Location(
   id number(8) PRIMARY KEY,
   city_id number(8) NOT NULL,
   street varchar2(30) NOT NULL,
   house_number number(4),
   postal_code varchar2(15),
   CONSTRAINT fk_location_city_id FOREIGN KEY (city_id)
   REFERENCES City (id)
);

CREATE TABLE Metal(
   id number(5) PRIMARY KEY,
   name varchar2(20) NOT NULL
);

CREATE TABLE Metallurgical_Plant(
   id number(5) PRIMARY KEY,
   mp_rate number(10),
```

```
    loc_id number(8),
    metal_id number(5) NOT NULL,
    CONSTRAINT fk_mplant_loc_id FOREIGN KEY (loc_id)
    REFERENCES Location (id),
    CONSTRAINT fk_mplant_metal_id FOREIGN KEY (metal_id)
    REFERENCES Metal (id)
);

CREATE TABLE Job(
    id number(5) PRIMARY KEY,
    title varchar2(40) NOT NULL,
    starting_salary number(10,2) NOT NULL,
    starting_abp number(4,1) NOT NULL
);

CREATE TABLE Employee(
    id number(8) PRIMARY KEY,
    name varchar2(30) NOT NULL,
    surname varchar2(30) NOT NULL,
    job_id number(5),
    salary number(10,2),
    ab_percentage number(4,1),
    annual_bonus number(10,2),
    plant_id number(5),
    CONSTRAINT fk_employee_job_id FOREIGN KEY (job_id)
```

```
    REFERENCES Job (id),
    CONSTRAINT fk_employee_plant_id FOREIGN KEY (plant_id)
    REFERENCES Metallurgical_Plant (id)
);

CREATE TABLE Ore(
    id number(8) PRIMARY KEY,
    name varchar2(30) NOT NULL,
    metal_concentration varchar2(20),
    metal_id number(5) NOT NULL,
    CONSTRAINT fk_ore_metal_id FOREIGN KEY (metal_id)
    REFERENCES Metal(id)
);

CREATE TABLE Supplier(
    id number(8) PRIMARY KEY,
    name varchar2(30) NOT NULL,
    phone_number varchar2(20),
    loc_id number(8),
    CONSTRAINT fk_supplier_loc_id FOREIGN KEY (loc_id)
    REFERENCES Location (id)
);

CREATE TABLE Supplier_Ore(
    supplier_id number(8),
```

```
    ore_id number(8),
    PRIMARY KEY (supplier_id, ore_id),
    CONSTRAINT fk_supplier_ore_id FOREIGN KEY (supplier_id)
    REFERENCES Supplier(id),
    CONSTRAINT fk_ore_supplier_id FOREIGN KEY (ore_id)
    REFERENCES Ore(id)
);

CREATE TABLE Customer(
    id number(8) PRIMARY KEY,
    name varchar2(30) NOT NULL,
    phone_number varchar2(20),
    loc_id number(8),
    CONSTRAINT fk_customer_loc_id FOREIGN KEY (loc_id)
    REFERENCES Location (id)
);

CREATE TABLE Contract(
    id number(10) PRIMARY KEY,
    customer_id number(8) NOT NULL,
    metal_id number(8) NOT NULL,
    weight_in_tons number(17,2) NOT NULL,
    price_per_ton number(10,2) NOT NULL,
    total_price number(30,2),
    CONSTRAINT fk_contract_customer_id FOREIGN KEY (customer_id)
```

```
    REFERENCES Customer (id),
    CONSTRAINT fk_contract_metal_id FOREIGN KEY (metal_id)
    REFERENCES Metal (id)
);
```

2. Druhá časť zadania

2.1. 2 pohľady (mám 3) s netriviálnym selektom nad jednou tabuľkou (nestačí použiť iba vymenovanie stĺpcov, treba použiť "niečo navyše" napr.: vstavané funkcie);

-- Tento pohľad zobrazuje každú špeciálnu zmluvu. Špeciálna zmluva je taká, kde rozdiel medzi cenou suroviny a skutočnou cenou je väčší ako 50 000. To znamená, že niektoré špeciálne detaily zmluvy cenu zvýšili alebo znížili.

-- Je to jeden z pohľadov s netriviálnym selektom. Používa vstavanu funkciu ABS.

CREATE OR REPLACE VIEW Special_Contracts AS

SELECT id, customer_id, metal_id, ABS((weight_in_tons * price_per_ton) - total_price) AS difference

FROM Contract

WHERE ABS((weight_in_tons * price_per_ton) - total_price) > 50000;

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Special_Contracts AS
2  SELECT id, customer_id, metal_id, ABS((weight_in_tons * price_per_ton) - total_price) AS difference
3  FROM Contract
4  WHERE ABS((weight_in_tons * price_per_ton) - total_price) > 50000;
5
6  select * from special_contracts;
```

```
View created.
```

ID	CUSTOMER_ID	METAL_ID	DIFFERENCE
1	1	6	800000
6	4	6	2400000
10	7	10	134210
16	10	1	149960

Download CSV

```
4 rows selected.
```

-- Tento pohľad zobrazuje každého zamestnanca, ktorý nemá prácu.

-- Je to jeden z pohľadov s netriviálnym selektom. Používa vstavanu funkciu CONCAT.

```
CREATE OR REPLACE VIEW Jobless_Employees AS
SELECT id, CONCAT(CONCAT(name, ' '), surname) AS full_name
FROM Employee
WHERE job_id IS NULL;
```

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Jobless_Employees AS
2  SELECT id, CONCAT(CONCAT(name, ' '), surname) AS full_name
3  FROM Employee
4  WHERE job_id IS NULL;
5
6  select * from jobless_employees;
```

```
View created.
```

ID	FULL_NAME
9	Liam Patel
10	Isabella Chen
19	Lilia Volchenko
20	Boris Mykhailenko

Download CSV

```
4 rows selected.
```

-- Tento pohľad zobrazuje každého zamestnanca, ktorý nemá žiadne informácie o závode.

-- Je to jeden z pohľadov s netriviálnym selektom. Používa vstavanu funkciu CONCAT.

```
CREATE OR REPLACE VIEW Plantless_Employees AS
SELECT id, CONCAT(CONCAT(name, ' '), surname) AS full_name
FROM Employee
WHERE plant_id IS NULL;
```

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Plantless_Employees AS
2  SELECT id, CONCAT(CONCAT(name, ' '), surname) AS full_name
3  FROM Employee
4  WHERE plant_id IS NULL;
5
6  select * from plantless_employees;
```

```
View created.
```

ID	FULL_NAME
4	Mia Johnson
5	Ahmed Khan
14	Sofia Szymanski
15	Yuri Radovanovic

Download CSV

```
4 rows selected.
```

2.2. 3 pohľady so spájaním tabuliek (1x spojenie 3 tabuliek, 1x outer join, 1x spojenie aspoň 2 tabuliek);

-- Tento pohľad zobrazuje podrobné informácie o každej zmluve. Dodatočne zobrazuje meno zákazníka zo tabuľky Customer a názov kovu z tabuľky Metal.

-- Pohľad používa spojenie 3 tabuliek.

CREATE OR REPLACE VIEW Detailed_Contract AS

SELECT ct.id, cr.name AS Customer_Name, cr.phone_number, m.name AS Metal, ct.weight_in_tons, ct.total_price

FROM Contract ct

JOIN Customer cr ON ct.customer_id = cr.id

JOIN Metal m ON ct.metal_id = m.id

ORDER BY cr.name;

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Detailed_Contract AS
2  SELECT ct.id, cr.name AS Customer_Name, cr.phone_number, m.name AS Metal, ct.weight_in_tons, ct.total_price
3  FROM Contract ct
4  JOIN Customer cr ON ct.customer_id = cr.id
5  JOIN Metal m ON ct.metal_id = m.id
6  ORDER BY cr.name;
7
8  select * from detailed_contract;
9
```

View created.

ID	CUSTOMER_NAME	PHONE_NUMBER	METAL	WEIGHT_IN_TONS	TOTAL_PRICE
7	ApexForge Industries	-	Silver	1600	-
16	AstroHorizon Innovations	-	Iron	62000	7523000
17	AstroHorizon Innovations	-	Zinc	1158	-
15	ElementalWorks Innovations	+49 176 9876 5432	Lithium	160	-
9	InnovateCrafter Corp.	-	Tin	7900	202030200
8	InnovateCrafter Corp.	-	Zinc	12000	-
4	QuantumPulse Labs	+44 20 3456 7890	Nickel	2300	40606000
3	QuantumPulse Labs	+44 20 3456 7890	Copper	4300	36747200
5	RoboLogic Solutions	+61 3 2109 8765	Aluminum	4500	-
11	StellarCraft Industries	+33 6 5432 1098	Mercury	326	-
10	StellarCraft Industries	+33 6 5432 1098	Platinum	5	220000000

-- Tento pohľad slúži na podrobné informácie o zamestnancoch. Dodatočne zobrazuje názov práce z tabuľky Job, názov mesta z tabuľky City, názov ulice z tabuľky Location, názov kovu, ktorý závod vyrába z tabuľky Metal.

-- Pohľad používa outer join.

CREATE OR REPLACE VIEW Detailed_Employee AS

SELECT e.id, e.name, e.surname, j.title AS Job_Title, c.name AS City, l.street AS Street, m.name AS Plant_Production, e.salary

FROM Employee e

LEFT OUTER JOIN Metallurgical_Plant mp ON e.plant_id = mp.id

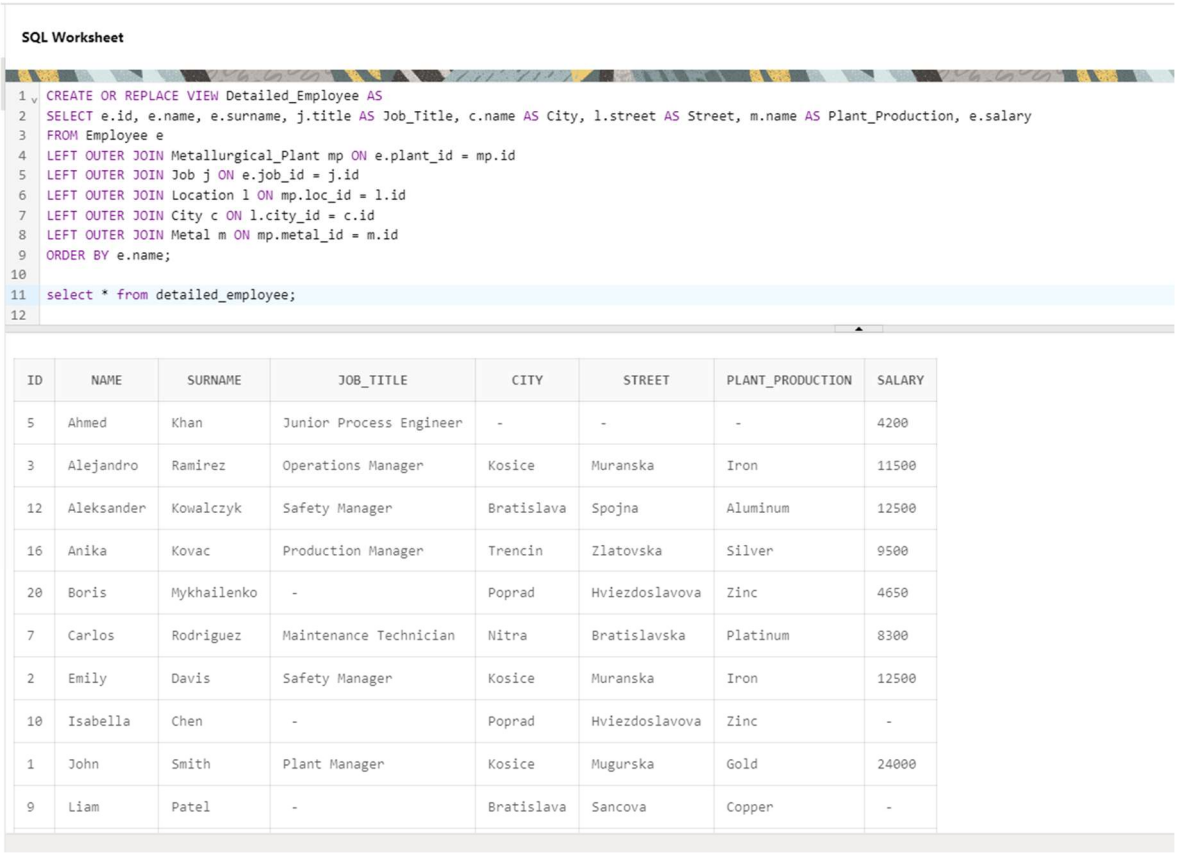
LEFT OUTER JOIN Job j ON e.job_id = j.id

LEFT OUTER JOIN Location l ON mp.loc_id = l.id

LEFT OUTER JOIN City c ON l.city_id = c.id

LEFT OUTER JOIN Metal m ON mp.metal_id = m.id

ORDER BY e.name;



SQL Worksheet

```
CREATE OR REPLACE VIEW Detailed_Employee AS
SELECT e.id, e.name, e.surname, j.title AS Job_Title, c.name AS City, l.street AS Street, m.name AS Plant_Production, e.salary
FROM Employee e
LEFT OUTER JOIN Metallurgical_Plant mp ON e.plant_id = mp.id
LEFT OUTER JOIN Job j ON e.job_id = j.id
LEFT OUTER JOIN Location l ON mp.loc_id = l.id
LEFT OUTER JOIN City c ON l.city_id = c.id
LEFT OUTER JOIN Metal m ON mp.metal_id = m.id
ORDER BY e.name;

select * from detailed_employee;
```

ID	NAME	SURNAME	JOB_TITLE	CITY	STREET	PLANT_PRODUCTION	SALARY
5	Ahmed	Khan	Junior Process Engineer	-	-	-	4200
3	Alejandro	Ramirez	Operations Manager	Kosice	Muranska	Iron	11500
12	Aleksander	Kowalczyk	Safety Manager	Bratislava	Spojna	Aluminum	12500
16	Anika	Kovac	Production Manager	Trencin	Zlatovska	Silver	9500
20	Boris	Mykhailenko	-	Poprad	Hviezdoslavova	Zinc	4650
7	Carlos	Rodriguez	Maintenance Technician	Nitra	Bratislavska	Platinum	8300
2	Emily	Davis	Safety Manager	Kosice	Muranska	Iron	12500
10	Isabella	Chen	-	Poprad	Hviezdoslavova	Zinc	-
1	John	Smith	Plant Manager	Kosice	Mugurska	Gold	24000
9	Liam	Patel	-	Bratislava	Sancova	Copper	-

-- Tento pohľad zobrazuje sortiment dodávateľov(Supplier). Aké rudy má každý dodávateľ a ich očakávaná koncentrácia kovu.

-- Zobrazuje názov dodávateľa z tabuľky Supplier, názov rudy z tabuľky Ore, názov kovu z tabuľky Metal, koncentráciu kovu z tabuľky Ore.

-- Pohľad používa join pre viac ako 2 tabuľky.

CREATE OR REPLACE VIEW Assortments AS

SELECT s.name AS Supplier, o.name AS Ore, m.name AS Contained_Metal, o.metal_concentration AS Expected_Concentration

FROM Supplier_Ore so

JOIN Ore o ON so.ore_id = o.id

JOIN Supplier s ON so.supplier_id = s.id

JOIN Metal m ON o.metal_id = m.id

ORDER BY o.name;

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Assortments AS
2  SELECT s.name AS Supplier, o.name AS Ore, m.name AS Contained_Metal, o.metal_concentration AS Expected_Concentration
3  FROM Supplier_Ore so
4  JOIN Ore o ON so.ore_id = o.id
5  JOIN Supplier s ON so.supplier_id = s.id
6  JOIN Metal m ON o.metal_id = m.id
7  ORDER BY o.name;
8
9  select * from assortments;
10
```

View created.

SUPPLIER	ORE	CONTAINED_METAL	EXPECTED_CONCENTRATION
Elemental Resources Co.	Anglesite	Lead	82%
OreLink Solutions	Anglesite	Lead	82%
Global Ore Nexus	Anglesite	Lead	82%
OreHarbor Ventures	Argentite	Silver	79%
PrimeOre Solutions	Argentite	Silver	79%
PrimeOre Solutions	Bornite	Copper	58%
Quantum Minerals Group	Böhmite	Aluminum	47%
TerraMetal Trading	Böhmite	Aluminum	47%
OreHarbor Ventures	Cassiterite	Tin	73%

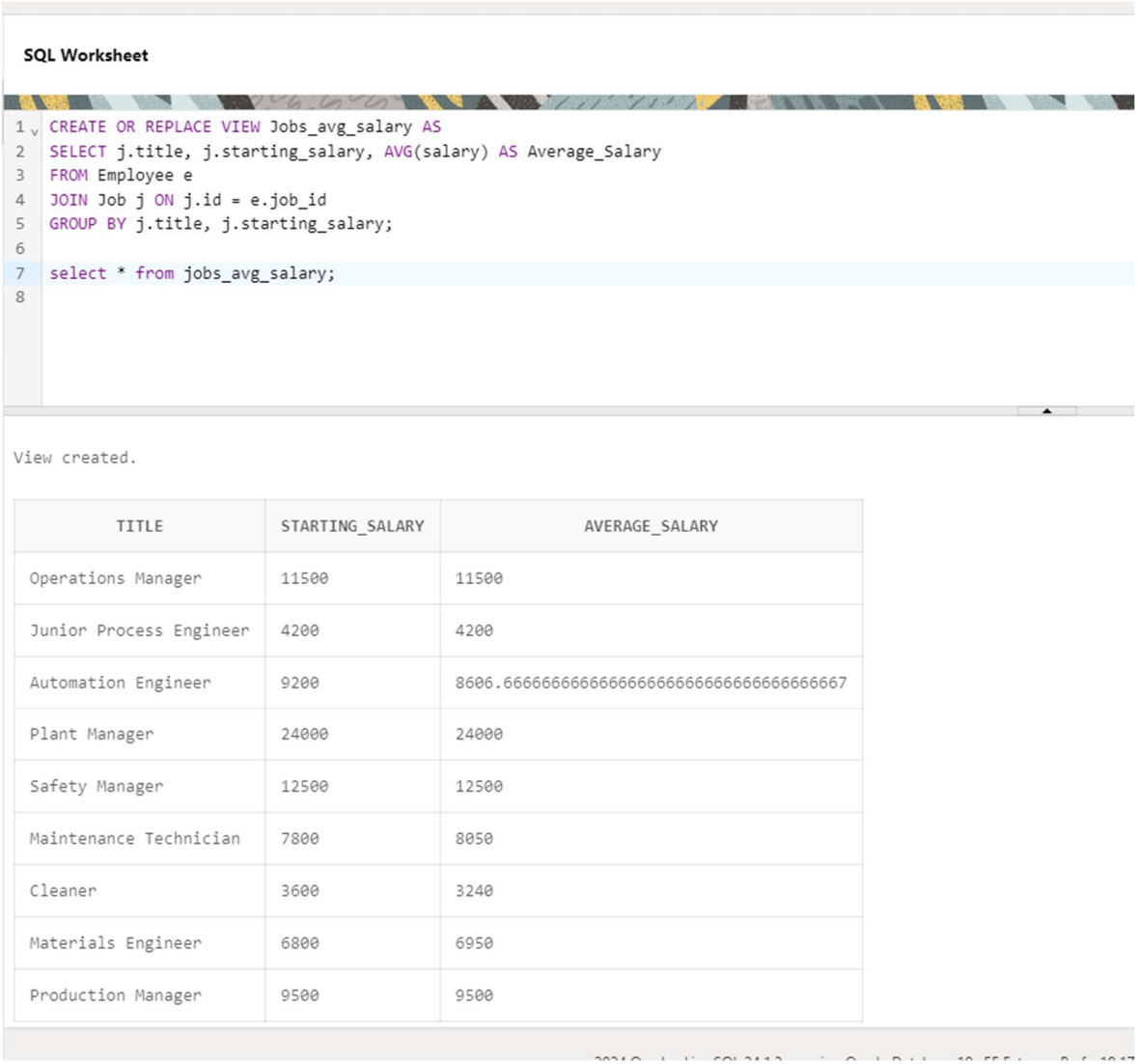
2.3. 2 pohľady s použitím agregačných funkcií alebo zoskupenia;

-- Tento pohľad zobrazuje priemernú mzdu pre každú prácu. Preberá informácie o skutočných platoch z tabuľky Employee.

-- Zobrazuje názov pracovnej pozície a nástupný plat z tabuľky Job a aktuálny priemerný plat na pozícii v spoločnosti.

-- Priemerná mzda sa vypočíta pomocou agregačnej funkcie AVG a pre každú prácu sa vypočíta pomocou GROUP BY s názvom pozície(title) a nástupným platom(starting_salary).

```
CREATE OR REPLACE VIEW Jobs_avg_salary AS
SELECT j.title, j.starting_salary, AVG(salary) AS Average_Salary
FROM Employee e
JOIN Job j ON j.id = e.job_id
GROUP BY j.title, j.starting_salary;
```



TITLE	STARTING_SALARY	AVERAGE_SALARY
Operations Manager	11500	11500
Junior Process Engineer	4200	4200
Automation Engineer	9200	8606.66666666666666666666666666666666666667
Plant Manager	24000	24000
Safety Manager	12500	12500
Maintenance Technician	7800	8050
Cleaner	3600	3240
Materials Engineer	6800	6950
Production Manager	9500	9500

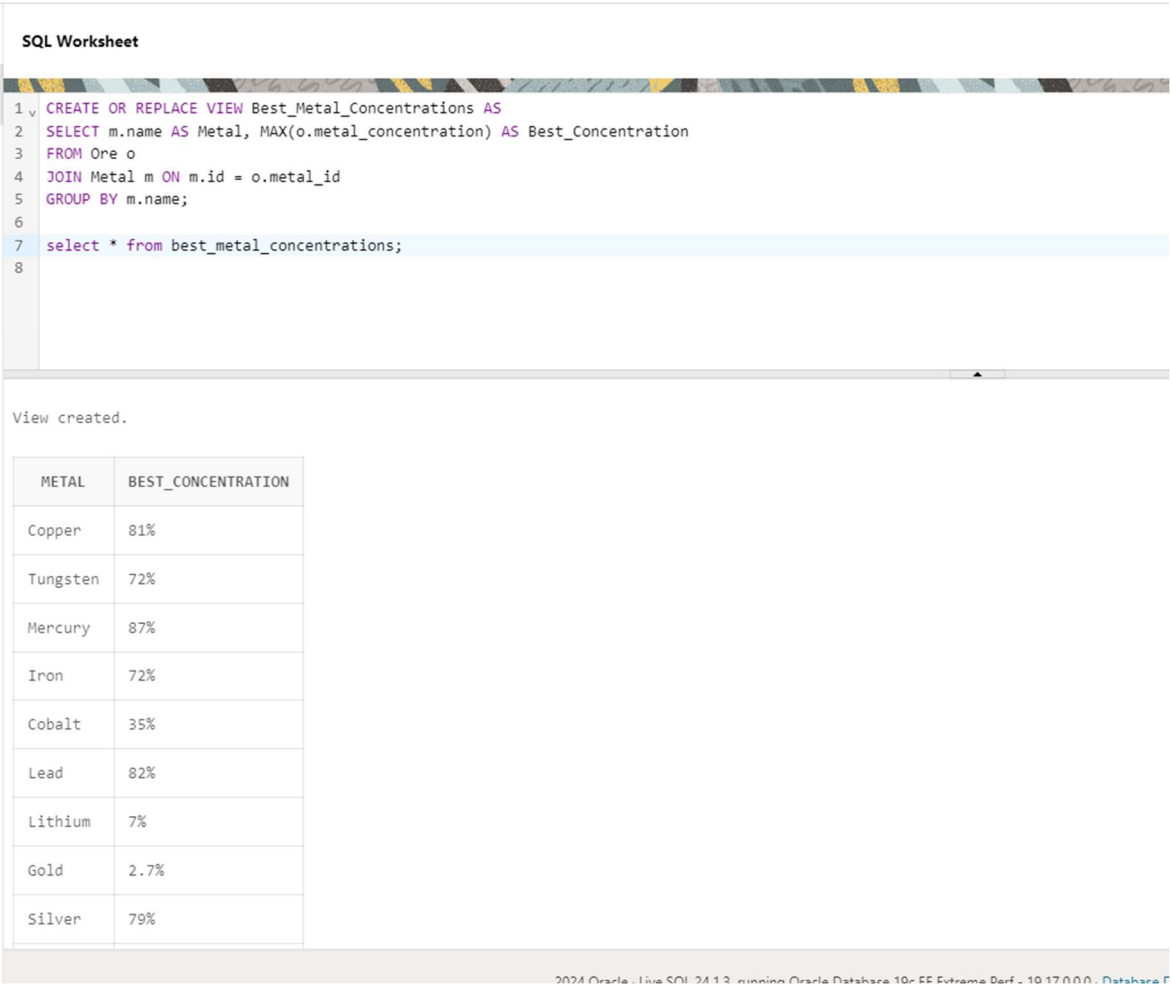
-- Tento pohľad zobrazuje najlepšiu koncentráciu kovu v rudách pre každý kov.

-- Zobrazuje názvy kovov z tabuľky Metal a najlepšiu koncentráciu kovu zistenú pomocou agregačnej funkcie MAX a GROUP BY pre názov kovu.

```
CREATE OR REPLACE VIEW Best_Metal_Concentrations AS
SELECT m.name AS Metal, MAX(o.metal_concentration) AS Best_Concentration
FROM Ore o
```

JOIN Metal m ON m.id = o.metal_id

GROUP BY m.name;



2.4. 1 pohľad (mám 2) s použitím množinových operácií;

-- Tento pohľad zobrazuje podrobné informácie o všetkých zariadeniach nachádzajúcich sa v Košiciach. Zariadeniami sú závody, zákazníci a dodávatelia.

-- Zobrazuje názov mesta z tabuľky City, typ zariadenia uvedený v každom združenom výbere samostatne, názov a telefónne číslo zariadenia z vlastných tabuliek (iba závody nemajú názov a telefónne číslo, takže majú null) a úplnú polohu (ulica a číslo domu z tabuľky Location).

-- Kosice sets

CREATE OR REPLACE VIEW Kosice_Facilities AS

SELECT c.name AS City, 'Plant' AS Type, NULL AS Name, NULL AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location

FROM Metallurgical_Plant mp

JOIN Location l ON mp.loc_id = l.id

```
JOIN City c ON l.city_id = c.id

WHERE c.name = 'Kosice'


UNION


SELECT c.name AS City, 'Supplier' AS Type, s.name AS Name, s.phone_number AS Phone_Number,
CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location

FROM Supplier s

JOIN Location l ON s.loc_id = l.id

JOIN City c ON l.city_id = c.id

WHERE c.name = 'Kosice'


UNION


SELECT c.name AS City, 'Customer' AS Type, cr.name AS Name, cr.phone_number AS
Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location

FROM Customer cr

JOIN Location l ON cr.loc_id = l.id

JOIN City c ON l.city_id = c.id

WHERE c.name = 'Kosice'

ORDER BY Type;
```

SQL Worksheet

```
CREATE OR REPLACE VIEW Kosice_Facilities AS
SELECT c.name AS City, 'Plant' AS Type, NULL AS Name, NULL AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location
FROM Metallurgical_Plant mp
JOIN Location l ON mp.loc_id = l.id
JOIN City c ON l.city_id = c.id
WHERE c.name = 'Kosice'

UNION

SELECT c.name AS City, 'Supplier' AS Type, s.name AS Name, s.phone_number AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location
FROM Supplier s
JOIN Location l ON s.loc_id = l.id
JOIN City c ON l.city_id = c.id
WHERE c.name = 'Kosice'

UNION

SELECT c.name AS City, 'Customer' AS Type, cr.name AS Name, cr.phone_number AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location
FROM Customer cr
JOIN Location l ON cr.loc_id = l.id
JOIN City c ON l.city_id = c.id
WHERE c.name = 'Kosice'
ORDER BY Type;

select * from kosice_facilities;
```

CITY	TYPE	NAME	PHONE_NUMBER	LOCATION
Kosice	Customer	RoboLogic Solutions	+61 3 2109 8765	Majova, 78
Kosice	Customer	TechSynth Innovations	+1 555-987-6543	Hlinkova, 21
Kosice	Plant	-	-	Mugurska, 2
Kosice	Plant	-	-	Muranska, 15

-- Tento pohľad zobrazuje podrobné informácie o všetkých zariadeniach nachádzajúcich sa v Bratislave. Zariadeniami sú závody, zákazníci a dodávatelia.

-- Zobrazuje názov mesta z tabuľky City, typ zariadenia uvedený v každom združenom výbere samostatne, názov a telefónne číslo zariadenia z vlastných tabuliek (iba závody nemajú názov a telefónne číslo, takže majú null) a úplnú polohu (ulica a číslo domu z tabuľky Location).

-- Bratislava set

CREATE OR REPLACE VIEW Bratislava_Facilities AS

SELECT c.name AS City, 'Plant' AS Type, NULL AS Name, NULL AS Phone_Number,
CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location

FROM Metallurgical_Plant mp

JOIN Location l ON mp.loc_id = l.id

JOIN City c ON l.city_id = c.id

WHERE c.name = 'Bratislava'

UNION

SELECT c.name AS City, 'Supplier' AS Type, s.name AS Name, s.phone_number AS Phone_Number,
CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location

FROM Supplier s

JOIN Location l ON s.loc_id = l.id

JOIN City c ON l.city_id = c.id

WHERE c.name = 'Bratislava'

UNION

SELECT c.name AS City, 'Customer' AS Type, cr.name AS Name, cr.phone_number AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location

FROM Customer cr

JOIN Location l ON cr.loc_id = l.id

JOIN City c ON l.city_id = c.id

WHERE c.name = 'Bratislava'

ORDER BY Type;

SQL Worksheet

```
CREATE OR REPLACE VIEW Bratislava_Facilities AS
SELECT c.name AS City, 'Plant' AS Type, NULL AS Name, NULL AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location
FROM Metallurgical_Plant mp
JOIN Location l ON mp.loc_id = l.id
JOIN City c ON l.city_id = c.id
WHERE c.name = 'Bratislava'

UNION

SELECT c.name AS City, 'Supplier' AS Type, s.name AS Name, s.phone_number AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location
FROM Supplier s
JOIN Location l ON s.loc_id = l.id
JOIN City c ON l.city_id = c.id
WHERE c.name = 'Bratislava'

UNION

SELECT c.name AS City, 'Customer' AS Type, cr.name AS Name, cr.phone_number AS Phone_Number, CONCAT(CONCAT(l.street, ', '), l.house_number) AS Location
FROM Customer cr
JOIN Location l ON cr.loc_id = l.id
JOIN City c ON l.city_id = c.id
WHERE c.name = 'Bratislava'
ORDER BY Type;

select * from bratislava_facilities;
```

CITY	TYPE	NAME	PHONE_NUMBER	LOCATION
Bratislava	Plant	-	-	Legionarska, 30
Bratislava	Plant	-	-	Sancova, 12
Bratislava	Plant	-	-	Spojna, 3

Download CSV

2.5. 2 pohľady (mám 3) s použitím netriviálnych vnorených selektov;

-- Tento pohľad zobrazuje informácie o zamestnancoch závodu, ktorý vyrába najväčší objem kovu.

```
CREATE OR REPLACE VIEW biggest_mp_rate_plant_employees AS
SELECT *
FROM Employee
WHERE plant_id =(
   SELECT id
        FROM Metallurgical_Plant
   WHERE mp_rate = (
        SELECT MAX(mp_rate)
        FROM Metallurgical_Plant
   )
);
```

SQL Worksheet

```
1  CREATE OR REPLACE VIEW biggest_mp_rate_plant_employees AS
2  SELECT *
3  FROM Employee
4  WHERE plant_id =(
5      SELECT id
6      FROM Metallurgical_Plant
7      WHERE mp_rate = (
8          SELECT MAX(mp_rate)
9          FROM Metallurgical_Plant
10     )
11 );
12
13 select * from biggest_mp_rate_plant_employees;
```

```
View created.
```

ID	NAME	SURNAME	JOB_ID	SALARY	AB_PERCENTAGE	ANNUAL_BONUS	PLANT_ID
2	Emily	Davis	3	12500	115	-	1
3	Alejandro	Ramirez	2	11500	110	-	1
19	Lilia	Volchenko	-	7300	-	-	1

Download CSV

```
3 rows selected.
```

-- Tento pohľad zobrazuje všetkých dodávateľov, ktorí majú vo svojom sortimente akúkoľvek železnú rudu.

```
CREATE OR REPLACE VIEW Iron_Ore_Suppliers AS
SELECT *
FROM Supplier
WHERE id IN (
        SELECT supplier_id
   FROM Supplier_Ore
   WHERE ore_id IN (
        SELECT id
        FROM Ore
        WHERE metal_id = (
                SELECT id
                FROM Metal
                Where name = 'Iron'
      )
   )
);
```

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Iron_Ore_Suppliers AS
2  SELECT *
3  FROM Supplier
4  WHERE id IN (
5      SELECT supplier_id
6      FROM Supplier_Ore
7      WHERE ore_id IN (
8          SELECT id
9          FROM Ore
10         WHERE metal_id = (
11             SELECT id
12             FROM Metal
13             Where name = 'Iron'
14         )
15     )
16 );
17
18 select * from iron_ore_suppliers;
```

View created.

ID	NAME	PHONE_NUMBER	LOC_ID
1	OreLink Solutions	+1 555-123-4567	3
2	Elemental Resources Co.	+44 20 7123 4567	4
3	OreHarbor Ventures	-	13
4	GeoMineral Supply	+33 1 8765 4321	14
5	OreMasters International	+61 2 9876 5432	15
7	TerraMetal Trading	-	17

-- Tento pohľad zobrazuje všetkých dodávateľov, ktorí majú vo svojom sortimente akúkoľvek niklovú rudu.

```
CREATE OR REPLACE VIEW Nickel_Ore_Suppliers AS
SELECT *
FROM Supplier
WHERE id IN (
        SELECT supplier_id
   FROM Supplier_Ore
   WHERE ore_id IN (
        SELECT id
        FROM Ore
        WHERE metal_id = (
                SELECT id
                FROM Metal
                Where name = 'Nickel'
      )
   )
);
```

SQL Worksheet

```
1  CREATE OR REPLACE VIEW Nickel_Ore_Suppliers AS
2  SELECT *
3  FROM Supplier
4  WHERE id IN (
5      SELECT supplier_id
6      FROM Supplier_Ore
7      WHERE ore_id IN (
8          SELECT id
9          FROM Ore
10         WHERE metal_id = (
11             SELECT id
12             FROM Metal
13             Where name = 'Nickel'
14         )
15     )
16 );
17
18 select * from nickel_ore_suppliers;
```

View created.

ID	NAME	PHONE_NUMBER	LOC_ID
1	OreLink Solutions	+1 555-123-4567	3
7	TerraMetal Trading	-	17
11	Global Ore Nexus	+55 11 98765 4321	-

2.6. vytvorte 1 sekvenciu na generovanie primárnych kľúčov a trigger, ktorý bude vkladať hodnoty do príslušných tabuliek;

```
-- Sekvencia na generovanie PK
CREATE SEQUENCE pk_generator
start with 1
increment by 1
nocycle;

-- Trigger na generovanie PK pre tabuľku Employee
CREATE OR REPLACE TRIGGER create_pk
	BEFORE INSERT OR UPDATE ON employee
	FOR EACH ROW
begin
	if :new.id is null then
		:new.id := pk_generator.nextval;
	end if;
end;
/

-- test vlozenim novych hodnot bez primary key
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Oliver', 'Smith', 1, 2);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Emily', 'Johnson', 3, 1);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('William', 'Williams', 2, 1);
INSERT INTO Employee (name, surname, job_id, salary) VALUES ('Charlotte', 'Brown', 12, 3240);
INSERT INTO Employee (name, surname, job_id) VALUES ('Jack', 'Taylor', 15);

Select * from employee;
```

ID	NAME	SURNAME	JOB_ID	SALARY	AB_PERCENTAGE	ANNUAL_BONUS	PLANT_ID
1	Oliver	Smith	1	-	-	-	2
2	Emily	Johnson	3	-	-	-	1
3	William	Williams	2	-	-	-	1
4	Charlotte	Brown	12	3240	-	-	-
5	Jack	Taylor	15	-	-	-	-

Download CSV

5 rows selected.

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

-- Trigger pridava zamestnancovi počiatočný plat a ab_percentage z tabuľky Job ak tieto hodnoty sú null

```
CREATE OR REPLACE TRIGGER insert_starting_salary
BEFORE INSERT ON employee
FOR EACH ROW
BEGIN
   IF :new.job_id IS NOT NULL THEN
      IF :new.salary IS NULL THEN
         SELECT starting_salary INTO :new.salary
         FROM job
         WHERE id = :new.job_id;
      END IF;

         IF :new.ab_percentage IS NULL THEN
            SELECT starting_abp INTO :new.ab_percentage
        FROM job
        WHERE id = :new.job_id;
          END IF;
```

```
        END IF;
END;
/

-- test vlozenim novych hodnot bez primary key
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('John', 'Smith', 1, 2);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Emily', 'Davis', 3, 1);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Alejandro', 'Ramirez', 2, 1);
INSERT INTO Employee (name, surname, job_id, salary) VALUES ('Mia', 'Johnson', 12, 3240);
INSERT INTO Employee (name, surname, job_id) VALUES ('Ahmed', 'Khan', 15);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Sophia', 'Kim', 8, 5);
INSERT INTO Employee (name, surname, job_id, salary, plant_id) VALUES ('Carlos', 'Rodriguez', 5,
8300, 8);
INSERT INTO Employee (name, surname, job_id, salary, plant_id) VALUES ('Olivia', 'Anderson', 8,
7420, 7);
INSERT INTO Employee (name, surname, plant_id) VALUES ('Liam', 'Patel', 3);
INSERT INTO Employee (name, surname, plant_id) VALUES ('Isabella', 'Chen', 6);

INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Milena', 'Novak', 1, 2);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Aleksander', 'Kowalczyk', 3, 4);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Stanislav', 'Dvorak', 2, 4);
INSERT INTO Employee (name, surname, job_id, salary) VALUES ('Sofia', 'Szymanski', 11, 6950);
INSERT INTO Employee (name, surname, job_id) VALUES ('Yuri', 'Radovanovic', 15);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Anika', 'Kovac', 4, 5);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Marek', 'Javanovic', 5, 8);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Radoslav', 'Dubrovsky', 8, 7);
INSERT INTO Employee (name, surname, salary, plant_id) VALUES ('Lilia', 'Volchenko', 7300, 1);
INSERT INTO Employee (name, surname, salary, plant_id) VALUES ('Boris', 'Mykhailenko', 4650, 6);

Select * from employee;
```

ID	NAME	SURNAME	JOB_ID	SALARY	AB_PERCENTAGE	ANNUAL_BONUS	PLANT_ID
1	John	Smith	1	24000	150	-	2
2	Emily	Davis	3	12500	115	-	1
3	Alejandro	Ramirez	2	11500	110	-	1
4	Mia	Johnson	12	3240	50	-	-
5	Ahmed	Khan	15	4200	60	-	-
6	Sophia	Kim	8	9200	90	-	5
7	Carlos	Rodriguez	5	8300	90	-	8
8	Olivia	Anderson	8	7420	90	-	7
9	Liam	Patel	-	-	-	-	3
10	Isabella	Chen	-	-	-	-	6
11	Milena	Novak	1	24000	150	-	2
12	Aleksander	Kowalczyk	3	12500	115	-	4
13	Stanislav	Dvorak	2	11500	110	-	4

3. Naplnenie vytvorenej schémy údajmi(prva časť zadania)

```
--City
INSERT INTO City (id, name) VALUES (1, 'Kosice');
INSERT INTO City (id, name) VALUES (2, 'Nitra');
INSERT INTO City (id, name) VALUES (3, 'Bratislava');
INSERT INTO City (id, name) VALUES (4, 'Zilina');
INSERT INTO City (id, name) VALUES (5, 'Presov');
INSERT INTO City (id, name) VALUES (6, 'Trnava');
INSERT INTO City (id, name) VALUES (7, 'Trencin');
INSERT INTO City (id, name) VALUES (8, 'Poprad');
INSERT INTO City (id, name) VALUES (9, 'Martin');
INSERT INTO City (id, name) VALUES (10, 'Zvolen');

--Location
INSERT INTO Location (id, city_id, street, house_number) VALUES (1, 1, 'Muranska', 15);
INSERT INTO Location (id, city_id, street, house_number) VALUES (2, 1, 'Mugurska', 2);
INSERT INTO Location (id, city_id, street, house_number) VALUES (3, 1, 'Kpt. Nalepku', 27);
INSERT INTO Location (id, city_id, street, house_number) VALUES (4, 1, 'Komeskeho', 125);
INSERT INTO Location (id, city_id, street, house_number) VALUES (5, 1, 'Hlinkova', 21);
INSERT INTO Location (id, city_id, street, house_number) VALUES (6, 1, 'Majova', 78);
INSERT INTO Location (id, city_id, street, house_number) VALUES (7, 2, 'Pieskova', 42);
INSERT INTO Location (id, city_id, street, house_number) VALUES (8, 2, 'Bratislavska', 176);
INSERT INTO Location (id, city_id, street, house_number) VALUES (9, 2, 'Jarocka', 18);
INSERT INTO Location (id, city_id, street, house_number) VALUES (10, 3, 'Legionarska', 30);
INSERT INTO Location (id, city_id, street, house_number) VALUES (11, 3, 'Sancova', 12);
INSERT INTO Location (id, city_id, street, house_number) VALUES (12, 3, 'Spojna', 3);
INSERT INTO Location (id, city_id, street, house_number) VALUES (13, 4, 'Halkova', 183);
INSERT INTO Location (id, city_id, street, house_number) VALUES (14, 4, 'Obchodna', 161);
INSERT INTO Location (id, city_id, street, house_number) VALUES (15, 4, 'Pod hajom', 21);
INSERT INTO Location (id, city_id, street, house_number) VALUES (16, 5, 'Metodova', 4);
INSERT INTO Location (id, city_id, street, house_number) VALUES (17, 5, 'Gorkeho', 9);
INSERT INTO Location (id, city_id, street, house_number) VALUES (18, 6, 'Hlboka', 13);
```

```
INSERT INTO Location (id, city_id, street, house_number) VALUES (19, 6, 'Coburgova', 25);

INSERT INTO Location (id, city_id, street, house_number) VALUES (20, 7, 'Elektricna', 6);

INSERT INTO Location (id, city_id, street, house_number) VALUES (21, 7, 'Zlatovska', 117);

INSERT INTO Location (id, city_id, street, house_number) VALUES (22, 8, 'Hviezdoslavova', 14);

INSERT INTO Location (id, city_id, street, house_number) VALUES (23, 8, 'Bernolakova', 17);

INSERT INTO Location (id, city_id, street, house_number) VALUES (24, 8, 'Vodarenska', 20);

INSERT INTO Location (id, city_id, street, house_number) VALUES (25, 9, 'Viliama Zingora', 122);

INSERT INTO Location (id, city_id, street, house_number) VALUES (26, 9, 'Mladeze', 26);

INSERT INTO Location (id, city_id, street, house_number) VALUES (27, 9, 'Polna', 29);

INSERT INTO Location (id, city_id, street, house_number) VALUES (28, 10, 'Jana Kalinciaka', 31);

INSERT INTO Location (id, city_id, street, house_number) VALUES (29, 10, 'Podbelova', 35);

INSERT INTO Location (id, city_id, street, house_number) VALUES (30, 10, 'Borovianska', 14);


--Job

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (1, 'Plant Manager', 24000, 150);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (2, 'Operations Manager', 11500,
110);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (3, 'Safety Manager', 12500, 115);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (4, 'Production Manager', 9500,
108);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (5, 'Maintenance Technician',
7800, 90);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (6, 'Metallurgical Engineer', 8100,
90);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (7, 'Process Engineer', 7500, 90);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (8, 'Automation Engineer', 9200,
90);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (9, 'Electrical Engineer', 7200, 85);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (10, 'Quality Control Inspector',
7500, 85);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (11, 'Materials Engineer', 6800,
80);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (12, 'Cleaner', 3600, 50);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (13, 'Quality Control Inspector
Assistant', 4600, 65);
```

```
INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (14, 'Junior Metallurgical Engineer', 4600, 65);

INSERT INTO Job (id, title, starting_salary, starting_abp) VALUES (15, 'Junior Process Engineer', 4200, 60);


--Metal

INSERT INTO Metal (id, name) VALUES (1, 'Iron');

INSERT INTO Metal (id, name) VALUES (2, 'Lead');

INSERT INTO Metal (id, name) VALUES (3, 'Copper');

INSERT INTO Metal (id, name) VALUES (4, 'Nickel');

INSERT INTO Metal (id, name) VALUES (5, 'Aluminum');

INSERT INTO Metal (id, name) VALUES (6, 'Gold');

INSERT INTO Metal (id, name) VALUES (7, 'Silver');

INSERT INTO Metal (id, name) VALUES (8, 'Zinc');

INSERT INTO Metal (id, name) VALUES (9, 'Tin');

INSERT INTO Metal (id, name) VALUES (10, 'Platinum');

INSERT INTO Metal (id, name) VALUES (11, 'Mercury');

INSERT INTO Metal (id, name) VALUES (12, 'Titanium');

INSERT INTO Metal (id, name) VALUES (13, 'Cobalt');

INSERT INTO Metal (id, name) VALUES (14, 'Tungsten');

INSERT INTO Metal (id, name) VALUES (15, 'Lithium');


--Metallurgical_Plant

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (1, 300000, 1, 1);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (2, 13000, 2, 6);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (3, 160000, 11, 3);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (4, 120000, 12, 5);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (5, 21000, 21, 7);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (6, 170000, 22, 8);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (7, 140000, 7, 9);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (8, 7000, 8, 10);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (9, 114000, 9, 4);

INSERT INTO Metallurgical_Plant (id, mp_rate, loc_id, metal_id) VALUES (10, 190000, 10, 2);
```

```
INSERT INTO Metallurgical_Plant (id, mp_rate, metal_id) VALUES (11, 17000, 11);
INSERT INTO Metallurgical_Plant (id, mp_rate, metal_id) VALUES (12, 77000, 12);
INSERT INTO Metallurgical_Plant (id, mp_rate, metal_id) VALUES (13, 92000, 13);
INSERT INTO Metallurgical_Plant (id, mp_rate, metal_id) VALUES (14, 64000, 14);
INSERT INTO Metallurgical_Plant (id, mp_rate, metal_id) VALUES (15, 35000, 15);

--Employee
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('John', 'Smith', 1, 2);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Emily', 'Davis', 3, 1);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Alejandro', 'Ramirez', 2, 1);
INSERT INTO Employee (name, surname, job_id, salary) VALUES ('Mia', 'Johnson', 12, 3240);
INSERT INTO Employee (name, surname, job_id) VALUES ('Ahmed', 'Khan', 15);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Sophia', 'Kim', 8, 5);
INSERT INTO Employee (name, surname, job_id, salary, plant_id) VALUES ('Carlos', 'Rodriguez', 5,
8300, 8);
INSERT INTO Employee (name, surname, job_id, salary, plant_id) VALUES ('Olivia', 'Anderson', 8,
7420, 7);
INSERT INTO Employee (name, surname, plant_id) VALUES ('Liam', 'Patel', 3);
INSERT INTO Employee (name, surname, plant_id) VALUES ('Isabella', 'Chen', 6);

INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Milena', 'Novak', 1, 2);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Aleksander', 'Kowalczyk', 3, 4);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Stanislav', 'Dvorak', 2, 4);
INSERT INTO Employee (name, surname, job_id, salary) VALUES ('Sofia', 'Szymanski', 11, 6950);
INSERT INTO Employee (name, surname, job_id) VALUES ('Yuri', 'Radovanovic', 15);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Anika', 'Kovac', 4, 5);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Marek', 'Javanovic', 5, 8);
INSERT INTO Employee (name, surname, job_id, plant_id) VALUES ('Radoslav', 'Dubrovsky', 8, 7);
INSERT INTO Employee (name, surname, salary, plant_id) VALUES ('Lilia', 'Volchenko', 7300, 1);
INSERT INTO Employee (name, surname, salary, plant_id) VALUES ('Boris', 'Mykhailenko', 4650, 6);

--Ore
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (1, 'Hematite', '66%', 1);
```

```
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (2, 'Magnetite', '72%', 1);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (3, 'Limonite', '60%', 1);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (4, 'Galena', '78%', 2);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (5, 'Anglesite', '82%', 2);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (6, 'Cerussite', '60%', 2);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (7, 'Chalcopyrite', '42%', 3);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (8, 'Bornite', '58%', 3);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (9, 'Chalcocite', '81%', 3);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (10, 'Pentlandite', '36%', 4);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (11, 'Nickeline', '52%', 4);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (12, 'Gibbsite', '52%', 5);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (13, 'Böhmite', '47%', 5);
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (14, 'Diaspore', '52%', 5);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (15, 'Free-Milling Gold Ore',
'2.7%', 6);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (16, 'Argentite', '79%', 7);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (17, 'Sphalerite', '68%', 8);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (18, 'Cassiterite', '73%', 9);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (19, 'Sperrylite', '63%', 10);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (20, 'Cinnabar', '87%', 11);

INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (21, 'Rutile', '92%', 12);
```

```
INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (22, 'Cobaltite', '35%', 13);


INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (23, 'Scheelite', '72%', 14);


INSERT INTO Ore (id, name, metal_concentration, metal_id) VALUES (24, 'Spodumene', '7%', 15);


--Supplier

INSERT INTO Supplier (id, name, phone_number, loc_id) VALUES (1, 'OreLink Solutions', '+1 555-123-4567', 3);

INSERT INTO Supplier (id, name, phone_number, loc_id) VALUES (2, 'Elemental Resources Co.', '+44 20 7123 4567', 4);

INSERT INTO Supplier (id, name, loc_id) VALUES (3, 'OreHarbor Ventures', 13);

INSERT INTO Supplier (id, name, phone_number, loc_id) VALUES (4, 'GeoMineral Supply', '+33 1 8765 4321', 14);

INSERT INTO Supplier (id, name, phone_number, loc_id) VALUES (5, 'OreMasters International', '+61 2 9876 5432', 15);

INSERT INTO Supplier (id, name, loc_id) VALUES (6, 'Apex Ore Providers', 16);

INSERT INTO Supplier (id, name, loc_id) VALUES (7, 'TerraMetal Trading', 17);

INSERT INTO Supplier (id, name, loc_id) VALUES (8, 'OreUnity Enterprises', 18);

INSERT INTO Supplier (id, name, phone_number, loc_id) VALUES (9, 'Quantum Minerals Group', '+49 30 9876 5432', 19);

INSERT INTO Supplier (id, name, loc_id) VALUES (10, 'PrimeOre Solutions', 20);

INSERT INTO Supplier (id, name, phone_number) VALUES (11, 'Global Ore Nexus', '+55 11 98765 4321');

INSERT INTO Supplier (id, name, phone_number) VALUES (12, 'OreSphere Industries', '+91 22 8765 4321');


--Supplier_Ore

INSERT INTO Supplier_Ore VALUES (1, 1);

INSERT INTO Supplier_Ore VALUES (1, 3);

INSERT INTO Supplier_Ore VALUES (1, 5);

INSERT INTO Supplier_Ore VALUES (1, 11);

INSERT INTO Supplier_Ore VALUES (2, 2);

INSERT INTO Supplier_Ore VALUES (2, 3);

INSERT INTO Supplier_Ore VALUES (2, 5);
```

```
INSERT INTO Supplier_Ore VALUES (2, 17);
INSERT INTO Supplier_Ore VALUES (3, 1);
INSERT INTO Supplier_Ore VALUES (3, 18);
INSERT INTO Supplier_Ore VALUES (3, 16);
INSERT INTO Supplier_Ore VALUES (4, 2);
INSERT INTO Supplier_Ore VALUES (4, 12);
INSERT INTO Supplier_Ore VALUES (5, 9);
INSERT INTO Supplier_Ore VALUES (5, 3);
INSERT INTO Supplier_Ore VALUES (6, 23);
INSERT INTO Supplier_Ore VALUES (6, 22);
INSERT INTO Supplier_Ore VALUES (7, 1);
INSERT INTO Supplier_Ore VALUES (7, 2);
INSERT INTO Supplier_Ore VALUES (7, 15);
INSERT INTO Supplier_Ore VALUES (7, 13);
INSERT INTO Supplier_Ore VALUES (8, 4);
INSERT INTO Supplier_Ore VALUES (9, 6);
INSERT INTO Supplier_Ore VALUES (9, 7);
INSERT INTO Supplier_Ore VALUES (10, 8);
INSERT INTO Supplier_Ore VALUES (10, 9);
INSERT INTO Supplier_Ore VALUES (11, 10);
INSERT INTO Supplier_Ore VALUES (11, 14);
INSERT INTO Supplier_Ore VALUES (12, 19);
INSERT INTO Supplier_Ore VALUES (12, 20);
INSERT INTO Supplier_Ore VALUES (12, 21);
INSERT INTO Supplier_Ore VALUES (12, 24);
INSERT INTO Supplier_Ore VALUES (7, 12);
INSERT INTO Supplier_Ore VALUES (7, 3);
INSERT INTO Supplier_Ore VALUES (7, 11);
INSERT INTO Supplier_Ore VALUES (7, 4);
INSERT INTO Supplier_Ore VALUES (8, 7);
INSERT INTO Supplier_Ore VALUES (9, 13);
INSERT INTO Supplier_Ore VALUES (9, 1);
INSERT INTO Supplier_Ore VALUES (10, 16);
```

```
INSERT INTO Supplier_Ore VALUES (10, 20);

INSERT INTO Supplier_Ore VALUES (11, 4);

INSERT INTO Supplier_Ore VALUES (11, 5);

INSERT INTO Supplier_Ore VALUES (12, 22);

INSERT INTO Supplier_Ore VALUES (12, 2);

INSERT INTO Supplier_Ore VALUES (12, 9);

INSERT INTO Supplier_Ore VALUES (12, 12);


--Customer

INSERT INTO Customer (id, name, phone_number, loc_id) VALUES (1, 'TechSynth Innovations', '+1 555-987-6543', 5);

INSERT INTO Customer (id, name, phone_number) VALUES (2, 'QuantumPulse Labs', '+44 20 3456 7890');

INSERT INTO Customer (id, name, phone_number, loc_id) VALUES (3, 'RoboLogic Solutions', '+61 3 2109 8765', 6);

INSERT INTO Customer (id, name, loc_id) VALUES (4, 'TitanAuto Dynamics', 24);

INSERT INTO Customer (id, name, loc_id) VALUES (5, 'ApexForge Industries', 25);

INSERT INTO Customer (id, name, loc_id) VALUES (6, 'InnovateCrafter Corp.', 26);

INSERT INTO Customer (id, name, phone_number) VALUES (7, 'StellarCraft Industries', '+33 6 5432 1098');

INSERT INTO Customer (id, name, phone_number) VALUES (8, 'VanguardPro Manufacturing', '+81 90 1234 5678');

INSERT INTO Customer (id, name, phone_number, loc_id) VALUES (9, 'ElementalWorks Innovations', '+49 176 9876 5432', 29);

INSERT INTO Customer (id, name, loc_id) VALUES (10, 'AstroHorizon Innovations', 30);



--Contract

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (1, 1, 6, 10, 56000000, 560800000);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (2, 1, 2, 5000, 2053, 10295000);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (3, 2, 3, 4300, 8546, 36747200);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (4, 2, 4, 2300, 17655, 40606000);
```

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (5, 3, 5, 4500, 2186.55);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (6, 4, 6, 12, 55200000, 660000000);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (7, 5, 7, 1600, 782892);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (8, 6, 8, 12000, 2520.42);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (9, 6, 9, 7900, 25574, 202030200);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (10, 7, 10, 5, 45826842, 229000000);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (11, 7, 11, 326, 8011);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (12, 8, 12, 585, 11250);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (13, 8, 13, 62, 28691.42, 1778842);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (14, 8, 14, 132, 282);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (15, 9, 15, 160, 37380);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton, total_price) VALUES (16, 10, 1, 62000, 118.92, 7523000);

INSERT INTO Contract (id, customer_id, metal_id, weight_in_tons, price_per_ton) VALUES (17, 10, 8, 1158, 2654.1);

Záver

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