Laboratorium IV

Wiktor Zmiendak

1. **Korzystając z operatorów SELECT, FROM i innych, napisz skrypty do tworzenia zapytań oraz podzapytań do tabel bazy danych o wybranym obszarze tematycznym:**

```
1  USE Airport;
2
3  SELECT * FROM plane;
4
5  SELECT age, salary FROM worker;
6
7  SELECT pasengers_slots, plane_id FROM plane;
8
9  SELECT parking_id, slots_count FROM parking WHERE slots_count > 100;
10
11 SELECT cafe_id, tables_count FROM cafe WHERE tables_count < 20;
```

Wypisujemy wszystkie informacje zawarte w tabeli plan

plane_id	company	weight	size	pasengers_slots	speed	id_runway
1	LOT	1000	50	10	600	NULL
111	EasyJet	1000	50	10	500	NULL
224	Ryaner	1000	50	103	500	NULL
667	EasyJet	3900	50	104	550	NULL
787	EasyJet	1000	50	102	550	NULL
832	EasyJet	9000	50	10	550	NULL
911	EasyJet	8000	50	102	600	NULL
1110	Swishair	200	50	10	750	NULL
4564	LOT	1000	50	200	500	NULL
5253	Ryaner	2500	50	200	500	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Wypisujemy informacje na temat wieku oraz wynagrodzenia pracowników

age	salary
60	3500
35	34500
60	3500
25	3500
35	3500
11	53500
25	3500
25	3500
60	3500
25	2500

Wypisujemy informacje na temat ilości miejsc pasażerskich dla poszczególnych id samolotów

pasengers_slots	plane_id
10	1
10	111
103	224
104	667
102	787
10	832
102	911
10	1110
200	4564
200	5253
NULL	NULL

Wypisuje informacje na temat ilości miejsc parkingowych dla poszczególnych id parkingów, gdzie ich ilość jest większa od 100

parking_id	slots_count
883	230
2985	200
3478	1000
7654	500
NULL	NULL

Wypisuje informacje na temat ilości stolików dla poszczególnych id cafe, gdzie ich ilość jest mniejsza od 20

cafe_id	tables_count
83	10
87	10
122	10
475	10
798	10
978	9
NULL	NULL

2. **Korzystając z operatorów ORDER BY napisz skrypty do sortowania danych podczas tworzenia zapytań do bazy danych z wybranego obszaru tematycznego:**

```sql
USE Airport;

SELECT *
FROM cafe
ORDER BY coffee_cost DESC;

SELECT *
FROM parking
WHERE slots_count > 200
ORDER BY slots_count;

SELECT *
FROM worker
WHERE worker_name = 'Janek'
ORDER BY salary;

SELECT runway_id, length, runway_condition
FROM runway
WHERE length > 200
ORDER BY runway_condition;

SELECT *
FROM parking
WHERE floor > 2
ORDER BY floor;
```

Wypisuje wszystkie informacje na temat cafe i segreguje je względem kosztu kawy malejąco

cafe_id	coffee_cost	tables_count
83	30	10
50	20	20
122	20	10
190	20	34
254	20	75
475	20	10
978	20	9
6009	20	60
798	16	10
87	10	10
NULL	NULL	NULL

Wypisuje wszystkie informacje na temat parkingu i segreguje je względem ilości miejsc, gdzie miejsc jest więcej niż 200

parking_id	slots_count	floor	open_hours	cost
883	230	1	10:00:00	45
7654	500	1	00:00:00	45
3478	1000	7	00:00:00	45
NULL	NULL	NULL	NULL	NULL

Wypisuje wszystkie informacje na temat pracowników i segreguje je względem zarobków dla osób o imieniu Janek

worker_id	worker_name	age	salary	working_hours	specialization	id_shop	id_cafe	id_parking	id_plane	id_gate
82643	Janek	25	2500	09:00:00	worker	NULL	NULL	NULL	NULL	NULL
173	Janek	60	3500	09:00:00	worker	NULL	NULL	NULL	NULL	NULL
563	Janek	60	3500	09:00:00	worker	NULL	NULL	NULL	NULL	NULL
628	Janek	25	3500	09:00:00	worker	NULL	NULL	NULL	NULL	NULL
757	Janek	35	3500	09:30:00	worker	NULL	NULL	NULL	NULL	NULL
9578	Janek	25	3500	09:00:00	worker	NULL	NULL	NULL	NULL	NULL
1052	Janek	11	53500	09:00:00	worker	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Wypisuje informacje o id pasa startowego, jego długości i kondycji, gdzie długość jest większa od 200. Dane są uszeregowane względem kondycji

runway_id	length	runway_condition
187	3000	A
3000	3000	A
6289	10000	A
9115	3000	A
54232	3000	A
2998	3000	B
4989	3000	C
762	3000	F
853	3000	F
NULL	NULL	NULL

Wypisuje wszystkie informacje na temat parkingów, gdzie ilość pięter jest większa od 2. Dane są uszeregowane względem ilości pięter

parking_id	slots_count	floor	open_hours	cost
781	100	3	09:30:00	45
647	100	4	00:00:00	45
1055	100	6	00:00:00	45
3478	1000	7	00:00:00	45
NULL	NULL	NULL	NULL	NULL

3. **Wykorzystując operatory IN, BETWEEN, LIKE, napisz skrypty do filtrowania danych podczas tworzenia zapytań do bazy danych z wybranego obszaru tematycznego:**

```
1  •  USE Airport;
2
3  •  SELECT *
4     FROM parking
5     WHERE floor BETWEEN 2 AND 6;
6
7  •  SELECT worker_id, salary
8     FROM worker
9     WHERE salary BETWEEN 5500 AND 100000;
10
11 •  SELECT plane_id, company
12    FROM plane
13    WHERE company in ('Swishair', 'Ryaner', 'LOT');
14
15 •  SELECT *
16    FROM cafe
17    WHERE coffee_cost LIKE 20;
18
19 •  SELECT *
20    FROM shop
21    WHERE open_hours LIKE '07:00:00';
```

Wypisuje wszystkie informacje na temat parkingów, gdzie ilość pięter jest między 2 a 6

parking_id	slots_count	floor	open_hours	cost
647	100	4	00:00:00	45
781	100	3	09:30:00	45
1055	100	6	00:00:00	45
2985	200	2	00:00:00	45
5684	100	2	08:00:00	45
NULL	NULL	NULL	NULL	NULL

Wypisuje informacje o zarobkach dla poszczególnych id pracowników, gdzie zarobki są z przedziału 5500 i 100000

worker_id	salary
353	34500
1052	53500
NULL	NULL

Wypisuje informacje o firmie oraz odpowiadającej jej id samolotu, gdzie nazwa firmy jest jedną z: (Swishair, Ryaner, LOT)

plane_id	company
1	LOT
224	Ryaner
1110	Swishair
4564	LOT
5253	Ryaner
NULL	NULL

Wypisuje wszystkie informacje na temat sklepów, gdzie godzina otwarcia wynosi 07:00:00

plane_id	company
1	LOT
224	Ryaner
1110	Swishair
4564	LOT
5253	Ryaner
NULL	NULL

4. **Korzystając z operatorów GROUP BY i HAVING, napisz skrypty grupujące wybrane dane podczas tworzenia zapytań analitycznych i sumarycznych do bazy danych wybranego obszaru tematycznego:**

```sql
USE Airport;

SELECT tables_count, coffee_cost
FROM cafe
GROUP BY tables_count, coffee_cost
HAVING coffee_cost > 10;

SELECT id_plane, gate_number
FROM gate
GROUP BY id_plane, gate_number
HAVING gate_number < 3;

SELECT runway_number, length
FROM runway
GROUP BY runway_number, length
HAVING length > 200 AND length < 10000;

SELECT parking_id, floor
FROM parking
GROUP BY parking_id, floor
HAVING floor = 1;

SELECT worker_name, age
FROM worker
GROUP BY worker_name, age
HAVING age > 30;
```

Wypisuje informacje o ilości stolików oraz koście kawy, gdzie kawa kosztuje więcej niż 10. Grupuje wszystko względem tych dwóch danych

tables_count	coffee_cost
20	20
10	30
10	20
34	20
75	20
10	16
9	20
60	20

Wypisuje informacje o numerze przejścia oraz kluczu obcym plane_id, gdzie numer przejścia jest większy od 2. Grupuje wszystko względem tych dwóch danych

id_plane	gate_number
NULL	1
NULL	2

Wypisuje informacje o numerze pasa startowego oraz jego długości, gdzie długość jest większa od 200 i mniejsza od 10000. Grupuje wszystko względem tych dwóch danych

runway_number	length
1	3000
7	3000
8	3000
2	3000
3	3000
4	3000
9	3000
5	3000

Wypisuje informacje o id parkingu i ilości pięter, gdzie ilość pięter wynosi 1. Grupuje wszystko względem tych dwóch danych

parking_id	floor
883	1
4484	1
7654	1
9098	1
NULL	NULL

Wypisuje informacje o imieniu pracownika i jego wieku, gdzie wiek jest większy od 30. Grupuje wszystko względem tych dwóch danych

worker_name	age
Janek	60
Michał	35
Janek	35
Pawel	60

5. **Korzystając z operatorów JOIN, INNERJOIN i OUTERJOIN, napisz skrypty generujące tabele przestawne z wybranych danych podczas tworzenia zapytań do bazy danych wybranego obszaru tematycznego:**

```
1  USE Airport;
2
3  SELECT plane.company, runway.runway_condition
4  FROM plane
5  INNER JOIN runway
6  ON plane.id_runway = runway.runway_id;
7
8  SELECT parking.open_hours, worker.working_hours
9  FROM parking
10 RIGHT JOIN worker
11 on worker.id_parking = parking.parking_id;
12
13 SELECT gate.gate_number, plane.company
14 FROM gate
15 RIGHT JOIN plane
16 on gate.id_plane = plane.plane_id;
17
18 SELECT worker.salary, gate.gate_number
19 FROM worker
20 LEFT JOIN gate
21 on worker.id_gate = gate.gate_id;
22
23 SELECT shop.product, worker.salary
24 FROM shop
25 RIGHT JOIN worker
26 on worker.id_shop = shop.shop_id;
```

Wypisuje firmę samolotu i łączę to ze stanem pasa startowego

company	runway_condition
LOT	A
EasyJet	A
Ryaner	B
EasyJet	A
EasyJet	F
EasyJet	F
EasyJet	A
Swishair	A
LOT	C
Ryaner	A

Wypisuje godziny otwarcia parkingów i łączę to z godzinami pracy pracowników

open_hours	working_hours
09:30:00	09:00:00
09:30:00	00:00:00
09:30:00	09:00:00
09:30:00	09:00:00
09:30:00	09:30:00
09:30:00	09:00:00
09:30:00	09:00:00
09:30:00	09:00:00
09:30:00	00:00:00
09:30:00	09:00:00

Wypisuje numer bramki i łączę to z nazwą firmy samolotu

gate_number	company
NULL	LOT
NULL	EasyJet
3	Ryaner
5	Ryaner
5	Ryaner
3	Ryaner
4	Ryaner
4	Ryaner
NULL	EasyJet
NULL	EasyJet
NULL	EasyJet

Wypisuje zarobki pracowników i łączę to z numerem bramki

salary	gate_number
3500	3
34500	2
3500	2
3500	2
3500	1
53500	1
3500	2
3500	1
3500	3
2500	1

Wypisuje produkty ze sklepów i łączę to z zarobkami pracowników

	product	salary
▶	tea	3500
	water	34500
	tea	3500
	tea	3500
	tea	3500
	water	53500
	water	3500
	water	3500
	water	3500
	water	2500

6. **Korzystając z funkcji agregujących oraz funkcji CASE, napisz skrypty tworzące podzapytania różnego typu do bazy danych wybranego obszaru tematycznego.**

```
1  USE Airport;
2
3  SELECT SUM(coffee_cost) as total_cost FROM cafe;
4  SELECT SUM(cost) as total_cost FROM parking;
5  SELECT MIN(age) as youngest FROM worker;
6
7  SELECT
8      CASE
9          WHEN coffee_cost > 20 THEN 'Expensive'
10         ELSE 'Affordable'
11     END AS if_expensive
12 FROM cafe;
13
14 SELECT
15     CASE
16         WHEN runway_condition = 'A' THEN 'Good'
17         ELSE 'BAD'
18     END AS if_condition
19 FROM runway;
20
```

Całkowita suma kosztów kawy

total_cost
196

Całkowita suma kosztów parkingów

total_cost
450

Najmłodszy pracownik

youngest
11

Sprawdza czy koszt jest dostateczny czy nie (poniżej 20)

if_expensive
Expensive
Affordable
Affordable
Affordable
Affordable
Affordable
Affordable
Affordable
Affordable

Sprawdza stan pasa startowego

if_condition
BAD
BAD
Good
BAD
Good
BAD
Good
Good
Good

7. **Korzystając z operatorów UNION, INTERSECT, EXCEPT, napisz skrypty do realizacji operacji mnogościowych algebry relacyjnej w zapytaniach próbkowania danych z tabel bazy danych z wybranego obszaru tematycznego.**

```
1  USE Airport;
2
3  SELECT company
4  FROM plane
5  UNION
6  SELECT product
7  FROM shop;
8
9  SELECT pasengers_slots
10 FROM plane
11 UNION
12 SELECT slots_count
13 FROM parking;
14
15 SELECT coffee_cost
16 FROM cafe
17 INTERSECT
18 SELECT cost
19 FROM parking;
20
21 SELECT open_hours
22 FROM parking
23 INTERSECT
24 SELECT working_hours
25 FROM worker;
26
27 SELECT floor
28 FROM parking
29 EXCEPT
30 SELECT gate_number
31 FROM gate;
```

company
LOT
EasyJet
Ryaner
Swishair
water
flowers
perfume
chocolate
tea
beer
tedy bear

	pasengers_slots
▶	10
	103
	104
	102
	200
	20
	100
	230
	1000
	500
	40

	coffee_cost

	open_hours
▶	00:00:00
	09:30:00

	floor
▶	6
	7

8. **Utwórz 2 widoki dla bazy danych z wybranego obszaru tematycznego.**

```
1  USE Airport;
2
3  DROP VIEW IF EXISTS RichWorkers;
4  CREATE VIEW RichWorkers AS
5  SELECT salary, worker_name
6  FROM worker
7  WHERE salary > 10000;
8
9  SELECT * FROM RichWorkers;
10
11 DROP VIEW IF EXISTS BadRunways;
12 CREATE VIEW BadRunways AS
13 SELECT *
14 FROM runway
15 WHERE runway_condition = 'C';
16
17 SELECT * FROM BadRunways;
```

Tworzymy widok z pracownikami zarabiającymi powyżej 10000

	salary	worker_name
▶	34500	Michał
	53500	Janek

Tworzymy widok z pasami startowymi w złej kondycji

	runway_id	runway_number	length	runway_condition
▶	4989	4	3000	C