

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

З екзаменаційної роботи з дисципліни
«Бази даних»

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 37

Виконав студент

ІП-24, Новиков Гліб Костянтинович
(шифр, прізвище, ім'я, по батькові)

24.01.2024

Київ 2024

Опис предметного середовища

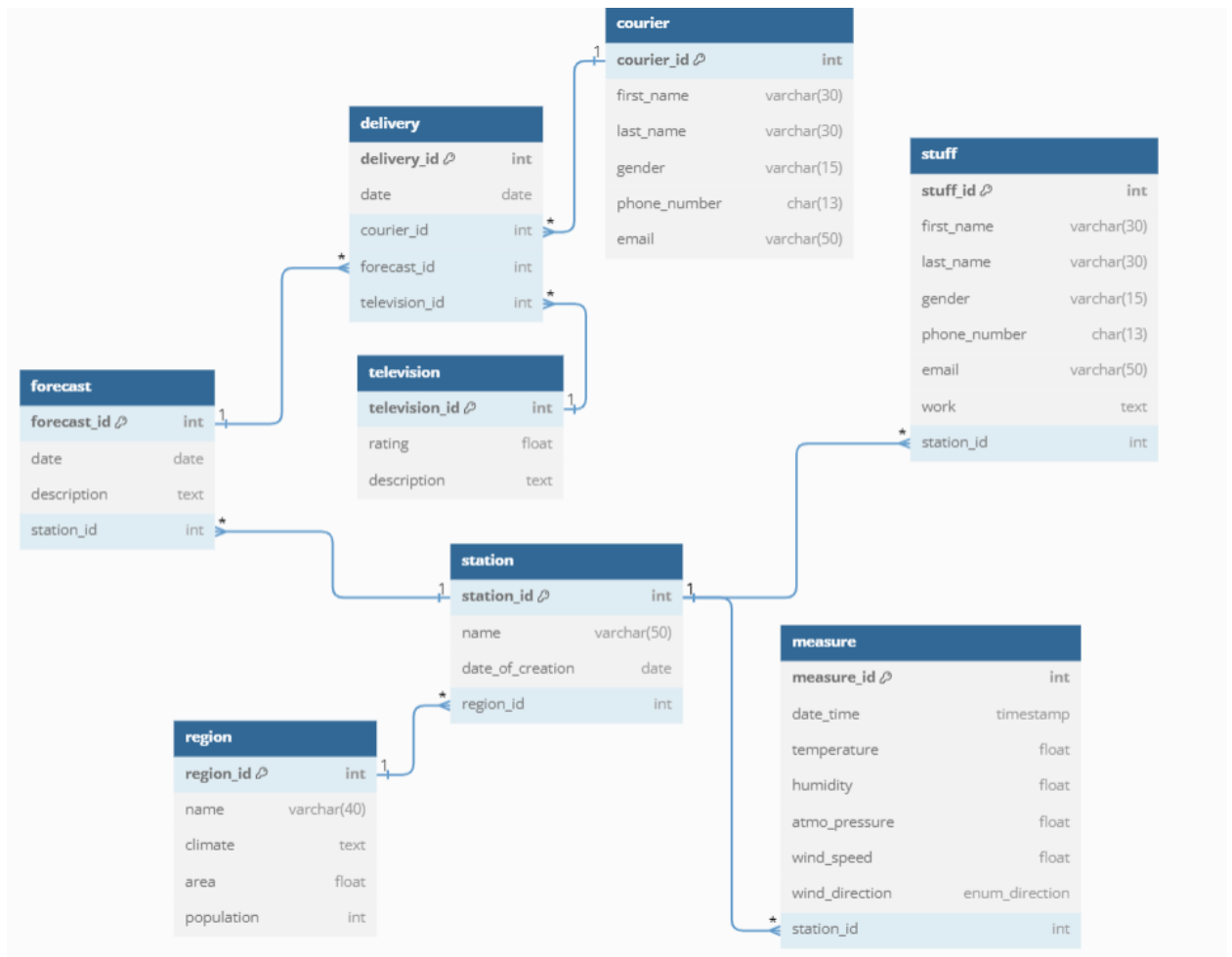
Метеорологічна служба забезпечує автоматичний контроль погодних характеристик. Серед них наступні: температура та вологість повітря, атмосферний тиск, швидкість на напрям вітру. Інформаційна система станції дає змогу визначити зміну температури та тиску, коефіцієнт різкості погодних умов. На основі вимірювань погодних параметрів складається прогноз погоди, який доставляється кур'єром на телебачення та радіо. Один й той самий кур'єр не може доставляти прогноз погоди на телебачення та радіо в один день.

Для заданого предметного середовища необхідно виконати наступне завдання:

- 1) Розробити ER-модель для заданого предметного середовища (мін. 5-6 сутностей). Відношення повинно знаходитись НФБК.
- 2) Згідно зі розробленої ER-моделі створити таблиці в БД засобами мови SQL. Передбачити необхідність наявності обмежень для підтримки посилальної цілісності та коректності значень, значень за замовченням. Для підтримки цілісності створити щонайменше 3 тригера AFTER <відповідна дія>.
- 3) Отримати звіт про прогноз погоди за минулий тиждень. У звіті в розрізі міст та дат відобразити погодні характеристики. По кожному місту вивести середньо тижневі показники. Для розв'язання поставленої задачі використати курсори та функції з параметрами та повертаємими значеннями.
- 4) Створити запити на основі їх текстового формулювання:
 - a) Найтепліший та найхолодніший місяці в м.Києві за останні 3 роки.
 - b) Кількість прогнозів, доставлених кур'єром на радіостанції, в назві котрих міститься «FM» за останній рік
 - c) Максимальна денна температура в липні місяці в м.Києві за 5 років від довільної дати.
 - d) Максимальний перепад денної температури (різниця температури за 2 дні) за останні 3 місяці в Київській області.

Обране СУБД: PosgreSQL.

Завдання 1



ER-модель інформаційної системи метеорологічної служби

Завдання 2

Створення спеціального типу для напрямку вітру:

```
create type enum_direction as enum (
```

```
    'схід', 'південь', 'південний захід', 'захід', 'північний захід', 'північ', 'північний
    схід', 'південний схід'
);
```

Створення таблиць бд:

```
create table region(
```

```
    region_id int generated always as identity primary key,
    name varchar(40) not null,
    climate text,
    area float,
```

```
population int  
);
```

```
create table station(  
station_id int generated always as identity primary key,  
name varchar(50) not null,  
date_of_creation date,  
region_id int not null,  
foreign key (region_id) references region(region_id)  
);
```

```
create table stuff(  
stuff_id int generated always as identity primary key,  
first_name varchar(30) not null,  
last_name varchar(30) not null,  
gender varchar(15),  
phone_number char(13),  
email varchar(50),  
work text,  
station_id int not null,  
foreign key (station_id) references station(station_id)  
);
```

```
create table measure(  
measure_id int generated always as identity primary key,  
date_time timestamp not null,  
temperature float,  
humidity float check (humidity >= 0 and humidity <= 100),  
atmo_pressure float,  
wind_speed float,
```

```
wind_direction enum_direction,  
station_id int not null,  
    foreign key (station_id) references station(station_id)  
);
```

```
create table forecast(  
    forecast_id int generated always as identity primary key,  
    date date not null,  
    description text not null,  
    station_id int not null,  
    foreign key (station_id) references station(station_id)  
);
```

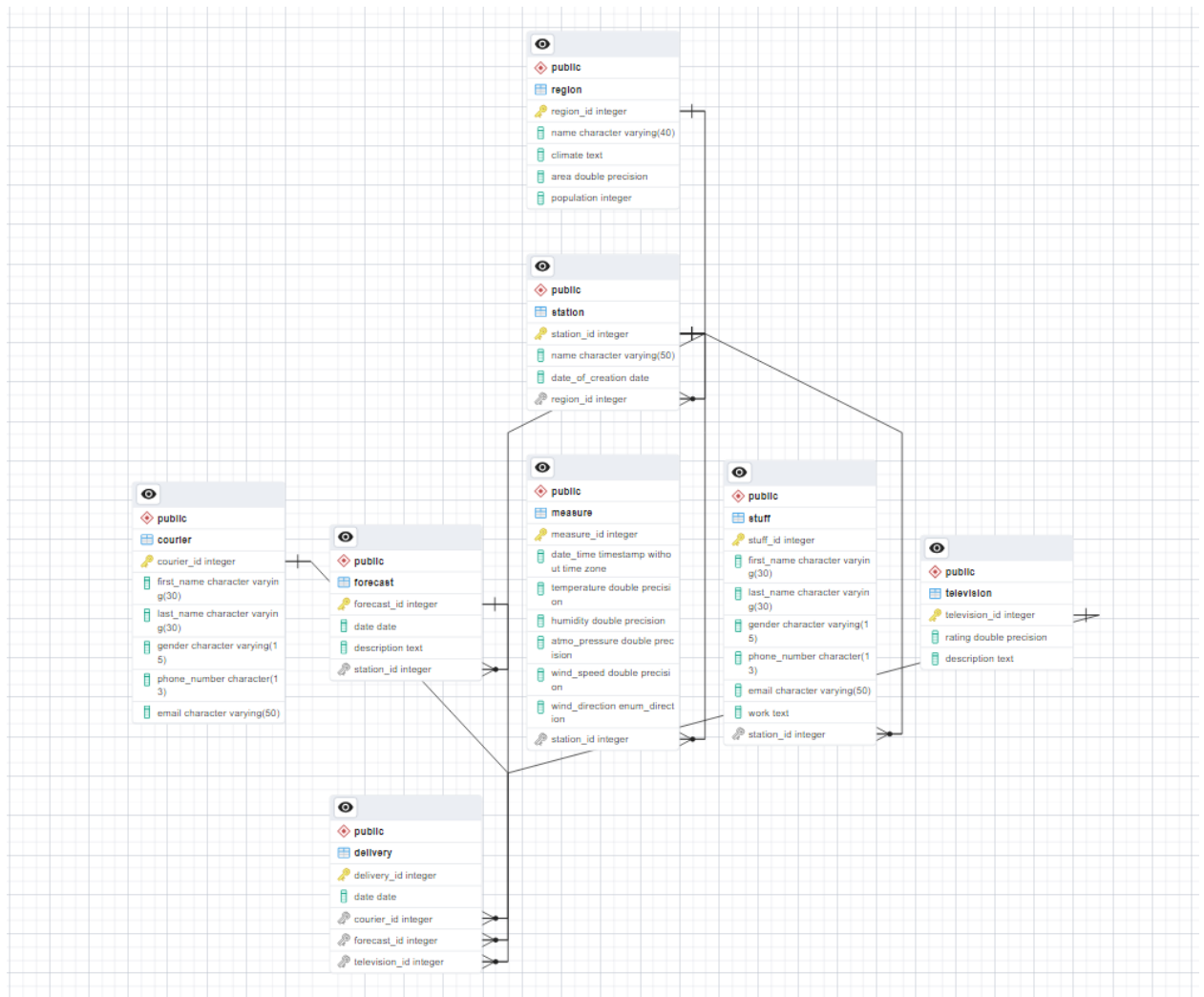
```
create table courier(  
    courier_id int generated always as identity primary key,  
    first_name varchar(30) not null,  
    last_name varchar(30) not null,  
    gender varchar(15),  
    phone_number char(13) not null,  
    email varchar(50)  
);
```

```
create table television(  
    television_id int generated always as identity primary key,  
    rating float check (rating >= 0 and rating <= 100),  
    description text  
);
```

```
create table delivery(  
    delivery_id int generated always as identity primary key,
```

date date,
 courier_id int not null,
 forecast_id int not null,
 television_id int not null,
 foreign key (courier_id) references courier(courier_id),
 foreign key (forecast_id) references forecast(forecast_id),
 foreign key (television_id) references television(television_id)
);

Реляційна модель бд, створена засобами субд PostgreSQL:



Створені тригери:

-- тригер для перевірки, щоб один кур'єр не доставляв декілька прогнозів

погоди на телебачення та радіо в один день (із опису предметного середовища).

```
CREATE OR REPLACE FUNCTION check_delivery()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
IF EXISTS (
```

```
SELECT 1
```

```
FROM delivery d
```

```
WHERE d.courier_id = NEW.courier_id
```

```
AND d.date = NEW.date
```

```
) THEN
```

```
RAISE EXCEPTION 'Цей кур`єр вже доставляв прогноз погоди на телебачення в цей день';
```

```
RETURN null;
```

```
END IF;
```

```
RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER delivery_check
```

```
BEFORE INSERT ON delivery
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION check_delivery();
```

```
ERROR:  Цей кур`єр вже доставляв прогноз погоди на телебачення в цей день  
CONTEXT:  Функція PL/pgSQL check_delivery() рядок 9 в RAISE
```

```
ПОМИЛКА:  Цей кур`єр вже доставляв прогноз погоди на телебачення в цей день  
SQL state: P0001
```

--Тригер для видалення вимірювань, прогнозів та співробітників при

видаленні станції

```
CREATE OR REPLACE FUNCTION delete_measurements()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM measure
    WHERE measure.station_id = OLD.station_id;
    DELETE FROM forecast
    WHERE forecast.station_id = OLD.station_id;
    DELETE FROM stuff
    WHERE stuff.station_id = OLD.station_id;
    RAISE NOTICE 'all measurings, forecast and stuff for this station deleted';
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER delete_measurements_trigger
before DELETE ON station
FOR EACH ROW
EXECUTE FUNCTION delete_measurements();
```

```
63 delete from station
64 where name = 'Sample Station';
```

Messages Data Output Notifications

ПОВІДОМЛЕННЯ: all measurings, forecast and stuff for this station deleted
DELETE 1

Query returned successfully in 53 msec.

--Тригер для автоматичного видалення застарілих вимірювань після додавання нового вимірювання

```
CREATE OR REPLACE FUNCTION delete_old_measurements()
RETURNS TRIGGER AS $$
```



```

BEGIN
DELETE FROM measure
WHERE date_time < (CURRENT_TIMESTAMP - INTERVAL '60 days');

RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER delete_old_measurements_trigger
AFTER INSERT ON measure
FOR EACH ROW
EXECUTE FUNCTION delete_old_measurements();

```

Завдання 3

Функція обрахує середні показники по регіонах від зазначеної дати до сьогоднішнього дня.

```

CREATE OR REPLACE FUNCTION generate_weather_report(start_date date)
RETURNS table(
    region_name varchar(40),
    avg_temperature float,
    avg_humidity float,
    avg_atmo_pressure float,
    avg_wind_speed float
) AS $$
DECLARE
    city_cursor CURSOR FOR
        SELECT name FROM region;

BEGIN
    return query select r.name, avg(temperature) avg_temperature,

```

```

        avg(humidity) avg_humidity, avg(atmo_pressure) avg_atmo_pressure,
avg(wind_speed) avg_wind_speed
    from measure
    join station using(station_id)
    join region r using(region_id)
    where date(date_time) > start_date
    group by r.name;

END;
$$ LANGUAGE plpgsql;

```

25 `select * from generate_weather_report('2024-01-17');`

Messages Data Output Notifications

Icons: [Menu] [Copy] [Paste] [Download] [Refresh] [Close]

	region_name character varying	avg_temperature double precision	avg_humidity double precision	avg_atmo_pressure double precision	avg_wind_speed double precision
1	Львів	19.2	60.333333333333336	1011.9666666666667	9.9
2	Одеса	18	70.1	1012.8	7.8
3	Харків	12.8	80.5	1014.6	5.2
4	Київ	17.2	63.7	1014.6	7.85

Завдання 4

4) Створити запити на основі їх текстового формулювання:

-- а) Найтепліший та найхолодніший місяці в м.Києві за останні 3 роки.

```

select EXTRACT(MONTH FROM date_time) _month, avg(temperature) avg_temp
from measure
join station using(station_id)
join region using(region_id)
group by EXTRACT(MONTH FROM date_time)
having avg(temperature) = (select max(avg_temp) from (select avg(temperature)
avg_temp from measure
join station using(station_id)

```

join region using(region_id)

group by EXTRACT(MONTH FROM date_time)))

or avg(temperature) = (select min(avg_temp) from (select
avg(temperature) avg_temp from measure

join station using(station_id)

join region using(region_id)

group by EXTRACT(MONTH FROM date_time));

	_month numeric	avg_temp double precision
1	1	16.29
2	2	-30

--b) Кількість прогнозів, доставлених кур'єром на радіостанції, в назві котрих міститься «FM» за останній рік

SELECT COUNT(d.delivery_id), c.first_name AS delivered_forecasts

FROM delivery d

JOIN forecast f ON d.forecast_id = f.forecast_id

JOIN station s ON f.station_id = s.station_id

JOIN courier c using(courier_id)

WHERE s.name LIKE '%FM%'

AND d.date >= current_date - INTERVAL '1 year'

group by c.first_name;

	count bigint		delivered_forecasts character varying (30)
1	3		Анна
2	2		Ірина
3	3		Олена
4	2		Максим
5	3		Володимир
6	2		Віталій
7	3		Олексій
8	3		Ігор
9	3		Наталія

--с) Максимальна денна температура в липні місяці в м.Києві за 5 років від довільної дати.

```
SELECT MAX(temperature) AS max_daily_temperature
FROM measure
JOIN station using(station_id)
join region using(region_id)
WHERE EXTRACT(MONTH FROM date_time) = 7
      AND EXTRACT(YEAR FROM date_time) BETWEEN EXTRACT(YEAR
FROM current_date) - 4 AND EXTRACT(YEAR FROM current_date)
      AND region.name = 'Київ';
```

26	--с) Максимальна денна температура в липні місяці в м.Києві за 5 років від довільної дати.
27	SELECT MAX(temperature) AS max_daily_temperature
28	FROM measure
29	JOIN station using(station_id)
30	join region using(region_id)
31	WHERE EXTRACT(MONTH FROM date_time) = 7
32	AND EXTRACT(YEAR FROM date_time) BETWEEN EXTRACT(YEAR FROM current_date) - 4 AND EXTRACT(YEAR FROM current_date)
33	AND region.name = 'Київ';

Messages	Data Output	Notifications
<div> <div>max_daily_temperature</div> <div>double precision</div> </div>		
1	30	

--d) Максимальний перепад денної температури (різниця температури за 2 дні) за останні 3 місяці в Київській області.

```
SELECT MAX(temperature_diff) AS max_temperature_difference
FROM (SELECT ABS(m1.temperature - m2.temperature) AS temperature_diff
```

```

FROM measure m1
JOIN measure m2 ON m1.station_id = m2.station_id
JOIN station s on m1.station_id = s.station_id
JOIN region using(region_id)
WHERE
    date(m1.date_time) = date(m2.date_time) - INTERVAL '2 days'
    AND m1.date_time >= current_date - INTERVAL '3 months'
    and region.name = 'Київ') AS temperature_diffs;

```

Результат запиту:

```

35  --d) Максимальний перепад денної температури (різниця температури за 2 д
36  SELECT MAX(temperature_diff) AS max_temperature_difference
37  FROM (SELECT ABS(m1.temperature - m2.temperature) AS temperature_diff
38         FROM measure m1
39         JOIN measure m2 ON m1.station_id = m2.station_id
40         JOIN station s on m1.station_id = s.station_id
41         JOIN region using(region_id)
42         WHERE
43             date(m1.date_time) = date(m2.date_time) - INTERVAL '2 days'
44             AND m1.date_time >= current_date - INTERVAL '3 months'
45             and region.name = 'Київ') AS temperature_diffs;

```

Messages **Data Output** Notifications

	max_temperature_difference double precision
1	14.299999999999999

Результат підзапиту:

```

36 SELECT MAX(temperature_diff) AS max_temperature_difference
37 FROM (SELECT ABS(m1.temperature - m2.temperature) AS temperature_diff
38       FROM measure m1
39       JOIN measure m2 ON m1.station_id = m2.station_id
40       JOIN station s ON m1.station_id = s.station_id
41       JOIN region using(region_id)
42       WHERE
43         date(m1.date_time) = date(m2.date_time) - INTERVAL '2 days'
44         AND m1.date_time >= current_date - INTERVAL '3 months'
45         and region.name = 'Київ') AS temperature_diffs;

```

Messages Data Output Notifications



	temperature_diff double precision
1	2.5
2	5.9000000000000002
3	2.1999999999999993
4	3.3999999999999986
5	3.3999999999999986
6	2.5
7	7.5
8	8.4000000000000002
9	8.4000000000000002
10	10.700000000000001

Додатково

Вибірка даних з усіх таблиць:

```

51 select * from courier;
52 select * from delivery;

```

Messages Data Output Notifications



	courier_id [PK] integer	first_name character varying (30)	last_name character varying (30)	gender character varying (15)	phone_number character	email character varying (50)
1	1	Олексій	Іванов	Чоловіча	+380951234567	oleksiy@example.com
2	2	Анна	Петренко	Жіноча	+380972345678	anna@example.com
3	3	Igor	Сидорчук	Чоловіча	+380503456789	igor@example.com
4	4	Наталія	Коваленко	Жіноча	+380964567890	natalia@example.com
5	5	Володимир	Мельник	Чоловіча	+380931234567	volodymyr@example.com
6	6	Олена	Шевченко	Жіноча	+380973456789	olena@example.com
7	7	Максим	Козак	Чоловіча	+380504567890	maxim@example.com
8	8	Ірина	Павленко	Жіноча	+380965678901	irina@example.com
9	9	Віталій	Григоренко	Чоловіча	+380936789012	vitaliy@example.com
10	10	Ольга	Литвин	Жіноча	+380978901234	olga@example.com

52 `select * from delivery;`

Messages Data Output Notifications

	delivery_id [PK] integer	date date	courier_id integer	forecast_id integer	television_id integer
1	1	2024-01-17	1	1	1
2	2	2024-01-17	2	2	2
3	3	2024-01-17	3	3	3
4	4	2024-01-18	4	4	4
5	5	2024-01-18	5	5	5
6	6	2024-01-18	6	6	1
7	7	2024-01-19	7	7	2
8	8	2024-01-19	8	8	3
9	9	2024-01-19	9	9	4
10	10	2024-01-20	1	10	5
11	11	2024-01-20	2	11	1
12	12	2024-01-20	3	12	2
Total rows: 24 of 24 Query complete 00:00:00.000					

53

select * from forecast;

Messages

Data Output

Notifications

	forecast_id [PK] integer	date date	description text	station_id integer
1	1	2023-01-01	Сонячно	1
2	2	2023-01-02	Хмарно з дощем	2
3	3	2023-01-03	Переважно ясно	3
4	4	2023-01-04	Туман	4
5	5	2023-01-05	Сильний вітер	5
6	6	2023-01-06	Мряка	6
7	7	2023-01-07	Легкий дощ	1
8	8	2023-01-08	Прохолодно	2
9	9	2023-01-09	Спекотно	3
10	10	2023-01-10	Сніг	4
11	11	2023-01-11	Заморозки	5
12	12	2023-01-12	Переважно хмарно	6
13	13	2023-01-13	Сонячно	1
14	14	2023-01-14	Дощ	2
15	15	2023-01-15	Метеорологічна нестабільність	3
16	16	2023-01-16	Спекотно	4
17	17	2023-01-17	Туман	5

54

select * from measure;

55

select * from region;

Messages

Data Output

Notifications

	measure_id [PK] integer	date_time timestamp without time zone	temperature double precision	humidity double precision	atmo_pressure double precision	wind_speed double precision	wind_direction enum_direction	station_id integer
1	1	2024-01-21 08:00:00	15.5	65.3	1015.2	8.2	схід	1
2	2	2024-01-22 12:00:00	20.3	58.7	1013.5	10.5	захід	2
3	3	2024-01-23 16:00:00	18	70.1	1012.8	7.8	північний схід	3
4	4	2024-01-24 20:00:00	12.8	80.5	1014.6	5.2	південний захід	4
5	5	2024-01-05 08:00:00	9.2	92.4	1011.3	3.7	південь	5
6	6	2024-01-06 12:00:00	24.1	45.8	1010.9	12.3	північний захід	6
7	7	2024-01-07 16:00:00	17.6	68.3	1013.7	9	схід	1
8	8	2024-01-08 20:00:00	13.4	75.9	1012.1	6.5	захід	2
9	9	2024-01-09 08:00:00	11.7	84.2	1014.8	4.8	північ	3
10	10	2024-01-10 12:00:00	26.5	38.6	1010.5	14.2	південний захід	4
11	11	2024-01-11 16:00:00	7.9	96.7	1009.2	2.3	південь	5
12	12	2024-01-12 20:00:00	19.8	55.4	1011.9	8.8	північний захід	6
13	13	2024-01-13 08:00:00	14.3	73.8	1013.4	6.1	схід	1
14	14	2024-01-14 12:00:00	22.7	49.2	1010.7	11	захід	2
15	15	2024-01-15 16:00:00	16.5	67	1012.3	8.3	північ	3


```
55 select * from region;
```

Messages Data Output Notifications

	region_id [PK] integer	name character varying (40)	climate text	area double precision	population integer
1	1	Київ	Континентальний	839.06	2884000
2	2	Львів	Континентальний	182.01	721301
3	3	Одеса	Морський	163.56	1008915
4	4	Харків	Континентальний	350.91	1441362

```
56 select * from station;
57 select * from stuff;
58 select * from television;
```

Messages Data Output Notifications

	station_id [PK] integer	name character varying (50)	date_of_creation date	region_id integer
1	1	Станція 1	2023-01-01	1
2	2	Станція 2	2023-02-15	2
3	3	Станція 3	2023-03-20	3
4	4	Станція 4	2023-04-10	4
5	5	Станція 5	2023-05-25	1
6	6	Станція 6	2023-06-30	2




```
57 select * from stuff;
58 select * from television;
```

Messages Data Output Notifications

	stuff_id [PK] integer	first_name character varying (30)	last_name character varying (30)	gender character varying (15)	phone_number character	email character varying (50)	work text	station_id integer
1	1	Іван	Петров	Чоловіча	+380951234567	ivan@example.com	Метеоролог	1
2	2	Олена	Іванова	Жіноча	+380972345678	olena@example.com	Аналітик	2
3	3	Андрій	Сидоренко	Чоловіча	+380503456789	andriy@example.com	Технік	3
4	4	Марія	Коваль	Жіноча	+380964567890	maria@example.com	Метеоролог	4
5	5	Петро	Морозов	Чоловіча	+380935678901	petro@example.com	Технік	5
6	6	Катерина	Шевченко	Жіноча	+380986789012	katerina@example.com	Аналітик	6
7	7	Ігор	Григоренко	Чоловіча	+380957890123	igor@example.com	Метеоролог	1
8	8	Наталя	Кузьменко	Жіноча	+380978901234	natalya@example.com	Технік	2
9	9	Сергій	Козак	Чоловіча	+380509012345	sergii@example.com	Аналітик	3
10	10	Юлія	Павленко	Жіноча	+380960123456	yulia@example.com	Метеоролог	4
11	11	Віталій	Литвин	Чоловіча	+380931234567	vitaliy@example.com	Аналітик	5
12	12	Ольга	Мельник	Жіноча	+380972345678	olga@example.com	Технік	6

```
58 select * from television;
59
```

Messages Data Output Notifications

	television_id [PK] integer 	rating double precision 	description text 
1	1	75.2	Новини 24/7
2	2	89.5	Музичний канал FM
3	3	65.8	Спортивний канал
4	4	92.1	Розважальний канал FM
5	5	50.3	Науковий канал