

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ «КВАДРА»

Описание функциональных характеристик

Дата: февраль, 2026 г.

Версия ПО: 1.5.0.5

ООО «Квадра системс»

https//www.quadra-system.ru

Аннотация

Настоящий документ содержит описание функциональных характеристик ПО СУБД «Квадра», в том числе описание ПО, информацию о назначении ПО, описание основных функциональных характеристик и возможностей ПО, а также задачи, реализуемые при помощи программы.

Глоссарий

Программа, ПО, СУБД	—	Программное обеспечение «Система управления базами данных «Квадра», разработки ООО «Квадра системс».
OpenJDK	—	Open Java Development Kit, набор инструментов и библиотек, позволяющих компилировать, выполнять и оптимизировать Java-приложения с открытым исходным кодом, распространяемый бесплатно.
Терминал, командная строка	—	Текстовый интерфейс, с помощью которого пользователь взаимодействует с операционной системой через интерфейс командной строки (CLI) или командную оболочку (shell).
JME	—	Java Management Extensions, стандартная технология Java для мониторинга и управления приложениями в реальном времени.
NoSQL	—	Not Only SQL, класс систем управления базами данных, не имеющих строгую реляционную модель, при этом имеющих горизонтальную масштабируемость, высокую производительность и гибкость.
P2P	—	Peer–to–peer архитектура системы с децентрализованным механизмом обмена информацией между узлами кластера, где каждый узел общается напрямую с другими узлами
CQL	—	Common Query Language, основной язык запросов для управления данными в СУБД «Квадра»

Содержание

1 ОПИСАНИЕ И НАЗНАЧЕНИЕ ПРОГРАММЫ 4

2 ВЫПОЛНЕНИЕ ПРОГРАММЫ 4

2.1 Сценарии работы с данными 4

2.1.1 Задача: Учет смен 4

2.1.2 Задача: Мониторинг параметров 9

2.1.3 Задача: Учет материалов на складе производства труб 11

2.2 Моделирование систем распределённой архитектуры 16

2.2.1 Задача: Создание однокластерной системы с двумя узлами данных 16

2.2.2 Задача: Добавление узла в кластер 17

3 КОНТАКТЫ 18

1 Описание и назначение программы

СУБД «Квадра» является серверным приложением, обеспечивающим взаимодействие с базами данных и данными, для систем распределенной архитектуры, требующих надежное хранение больших массивов информации, высокую доступность и масштабирование.

Программа представляет собой высокопроизводительную распределённую систему управления базами данных класса NoSQL.

СУБД «Квадра» позволяет осуществлять:

- моделирование (создание) систем распределённой архитектуры (P2P);
- управление элементами систем распределённой архитектуры (кластерами, узлами хранения данных и данными на этих узлах);
- реализацию различных сценариев работы с данным.

Системы, созданные посредствам СУБД «Квадра», обладают следующими преимуществами:

- имеют высокую скорость записи данных;
- обладают высокой доступностью и отказоустойчивостью;
- хранят большие объемы данных;
- позволяют линейно масштабироваться до, практически неограниченных размеров;
- имеют автоматическую репликацию между узлами и кластерами.

2 Выполнение программы

2.1 Сценарии работы с данными

2.1.1 Задача: Учет смен

Решение задачи «Учет смен» позволяет фиксировать во сколько приходит/уходит специалист, а также сколько времени он находится на рабочем месте.

Все приведённые команды выполняются в терминале утилиты cqlsh:

```
#создаем базу данных
CREATE KEYSPACE refinery_personnel
WITH replication = {
  'class': 'NetworkTopologyStrategy',
```

```
  'datacenter1': 1
};

#подключаемся к созданной базе данных, чтобы с ней взаимодействовать
USE refinery_personnel;

#создаем таблицу
CREATE TABLE personnel_entries (
    employee_id INT,
    entry_date DATE,
    entry_time TIMESTAMP,
    exit_time TIMESTAMP,
    employee_name TEXT,
    department TEXT,
    shift TEXT,
    access_gate TEXT,
    purpose TEXT,
    badge_id BIGINT,
    is_present BOOLEAN,
    PRIMARY KEY (employee_id, entry_date, entry_time)
) WITH CLUSTERING ORDER BY (entry_date DESC, entry_time DESC);

#создаем таблицу
CREATE TABLE current_presence (
    employee_id INT PRIMARY KEY,
    employee_name TEXT,
    department TEXT,
    shift TEXT,
    entry_time TIMESTAMP,
    access_gate TEXT,
    entry_timestamp BIGINT,
    last_updated TIMESTAMP
);

#создаем таблицу
CREATE TABLE presence_by_shift (
    shift TEXT,
    employee_id INT,
    employee_name TEXT,
    department TEXT,
    entry_time TIMESTAMP,
    access_gate TEXT,
    PRIMARY KEY (shift, employee_id)
);

#создаем таблицу
CREATE TABLE access_log (
    access_gate TEXT,
    access_date DATE,
    access_time TIMESTAMP,
    employee_id INT,
    employee_name TEXT,
    action TEXT,
    badge_id BIGINT,
    PRIMARY KEY (access_gate, access_date, access_time)
) WITH CLUSTERING ORDER BY (access_date DESC, access_time DESC);

#вносим данные в таблицу
INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department,
shift, access_gate, purpose, badge_id, is_present)
VALUES (1001, '2026-03-23', '2026-03-23 07:45:00', null, 'Иван Петров', 'Оперативный цех', 'day', 'КПП-
Северный', 'работа', 12345, true);

INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department,
shift, access_gate, purpose, badge_id, is_present)
VALUES (1002, '2026-03-23', '2026-03-23 07:50:00', null, 'Сергей Иванов', 'Цех переработки', 'day', 'КПП-
Восточный', 'работа', 12346, true);

INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department,
shift, access_gate, purpose, badge_id, is_present)
VALUES (1003, '2026-03-23', '2026-03-23 19:45:00', null, 'Алексей Смирнов', 'Оперативный цех', 'night',
'КПП-Северный', 'работа', 12347, true);
```

```
INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department, shift, access_gate, purpose, badge_id, is_present)
VALUES (1004, '2026-03-23', '2026-03-23 08:15:00', null, 'Михаил Волков', 'Лаборатория', 'day', 'КПП-Западный', 'работа', 12348, true);

INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department, shift, access_gate, purpose, badge_id, is_present)
VALUES (1005, '2026-03-23', '2026-03-23 09:30:00', null, 'Андрей Новиков', 'Ремонтная служба', 'flexible', 'КПП-Восточный', 'работа', 12349, true);

INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department, shift, access_gate, purpose, badge_id, is_present)
VALUES (1001, '2026-03-22', '2026-03-22 07:30:00', '2026-03-22 19:30:00', 'Иван Петров', 'Оперативный цех', 'day', 'КПП-Северный', 'работа', 12345, false);

INSERT INTO personnel_entries (employee_id, entry_date, entry_time, exit_time, employee_name, department, shift, access_gate, purpose, badge_id, is_present)
VALUES (1002, '2026-03-22', '2026-03-22 07:45:00', '2026-03-22 19:45:00', 'Сергей Иванов', 'Цех переработки', 'day', 'КПП-Восточный', 'работа', 12346, false);

INSERT INTO current_presence (employee_id, employee_name, department, shift, entry_time, access_gate, entry_timestamp, last_updated)
VALUES (1001, 'Иван Петров', 'Оперативный цех', 'day', '2026-03-23 07:45:00', 'КПП-Северный', 1742712300, toTimestamp(now()));

INSERT INTO current_presence (employee_id, employee_name, department, shift, entry_time, access_gate, entry_timestamp, last_updated)
VALUES (1002, 'Сергей Иванов', 'Цех переработки', 'day', '2026-03-23 07:50:00', 'КПП-Восточный', 1742712600, toTimestamp(now()));

INSERT INTO current_presence (employee_id, employee_name, department, shift, entry_time, access_gate, entry_timestamp, last_updated)
VALUES (1003, 'Алексей Смирнов', 'Оперативный цех', 'night', '2026-03-23 19:45:00', 'КПП-Северный', 1742751900, toTimestamp(now()));

INSERT INTO current_presence (employee_id, employee_name, department, shift, entry_time, access_gate, entry_timestamp, last_updated)
VALUES (1004, 'Михаил Волков', 'Лаборатория', 'day', '2026-03-23 08:15:00', 'КПП-Западный', 1742714100, toTimestamp(now()));

INSERT INTO current_presence (employee_id, employee_name, department, shift, entry_time, access_gate, entry_timestamp, last_updated)
VALUES (1005, 'Андрей Новиков', 'Ремонтная служба', 'flexible', '2026-03-23 09:30:00', 'КПП-Восточный', 1742718600, toTimestamp(now()));

INSERT INTO presence_by_shift (shift, employee_id, employee_name, department, entry_time, access_gate)
VALUES ('day', 1001, 'Иван Петров', 'Оперативный цех', '2026-03-23 07:45:00', 'КПП-Северный');

INSERT INTO presence_by_shift (shift, employee_id, employee_name, department, entry_time, access_gate)
VALUES ('day', 1002, 'Сергей Иванов', 'Цех переработки', '2026-03-23 07:50:00', 'КПП-Восточный');

INSERT INTO presence_by_shift (shift, employee_id, employee_name, department, entry_time, access_gate)
VALUES ('night', 1003, 'Алексей Смирнов', 'Оперативный цех', '2026-03-23 19:45:00', 'КПП-Северный');

INSERT INTO presence_by_shift (shift, employee_id, employee_name, department, entry_time, access_gate)
VALUES ('day', 1004, 'Михаил Волков', 'Лаборатория', '2026-03-23 08:15:00', 'КПП-Западный');

INSERT INTO presence_by_shift (shift, employee_id, employee_name, department, entry_time, access_gate)
VALUES ('flexible', 1005, 'Андрей Новиков', 'Ремонтная служба', '2026-03-23 09:30:00', 'КПП-Восточный');

INSERT INTO access_log (access_gate, access_date, access_time, employee_id, employee_name, action, badge_id)
VALUES ('КПП-Северный', '2026-03-23', '2026-03-23 07:45:00', 1001, 'Иван Петров', 'entry', 12345);

INSERT INTO access_log (access_gate, access_date, access_time, employee_id, employee_name, action, badge_id)
VALUES ('КПП-Восточный', '2026-03-23', '2026-03-23 07:50:00', 1002, 'Сергей Иванов', 'entry', 12346);

INSERT INTO access_log (access_gate, access_date, access_time, employee_id, employee_name, action, badge_id)
VALUES ('КПП-Северный', '2026-03-23', '2026-03-23 19:45:00', 1003, 'Алексей Смирнов', 'entry', 12347);

INSERT INTO access_log (access_gate, access_date, access_time, employee_id, employee_name, action, badge_id)
VALUES ('КПП-Западный', '2026-03-23', '2026-03-23 08:15:00', 1004, 'Михаил Волков', 'entry', 12348);
```

```
INSERT INTO access_log (access_gate, access_date, access_time, employee_id, employee_name, action, badge_id)
VALUES ('КПП-Восточный', '2026-03-23', '2026-03-23 09:30:00', 1005, 'Андрей Новиков', 'entry', 12349);

#проверяем заполнение таблиц
SELECT * FROM current_presence;
```

```
cqlsh:refinery_personnel> SELECT * FROM current_presence;

 employee_id | access_gate    | department       | employee_name   | entry_time                      | entry_timestamp | last_updated                    | shift
-------------+----------------+------------------+-----------------+---------------------------------+-----------------+---------------------------------+----------
        1004 | КПП-Западный   |      Лаборатория |   Михаил Волков | 2026-03-23 05:15:00.000000+0000 |      1742714100 | 2026-03-26 20:26:07.922000+0000 |      day
        1005 | КПП-Восточный | Ремонтная служба |  Андрей Новиков | 2026-03-23 06:30:00.000000+0000 |      1742718600 | 2026-03-26 20:26:07.931000+0000 | flexible
        1001 |  КПП-Северный |  Оперативный цех |     Иван Петров | 2026-03-23 04:45:00.000000+0000 |      1742712300 | 2026-03-26 20:26:07.887000+0000 |      day
        1003 |  КПП-Северный |  Оперативный цех | Алексей Смирнов | 2026-03-23 16:45:00.000000+0000 |      1742751900 | 2026-03-26 20:26:07.913000+0000 |    night
        1002 | КПП-Восточный |  Цех переработки |   Сергей Иванов | 2026-03-23 04:50:00.000000+0000 |      1742712600 | 2026-03-26 20:26:07.894000+0000 |      day
```

```
SELECT * FROM personnel_entries;
```

```
cqlsh:refinery_personnel> SELECT * FROM personnel_entries;

 employee_id | entry_date | entry_time                      | access_gate    | badge_id | department       | employee_name   | exit_time                       | is_present |
 purpose | shift
-------------+------------+---------------------------------+----------------+----------+------------------+-----------------+---------------------------------+------------+
---------+----------
        1004 | 2026-03-23 | 2026-03-23 05:15:00.000000+0000 |  КПП-Западный |    12348 |      Лаборатория |   Михаил Волков |                            null |       True |
  работа |      day
        1005 | 2026-03-23 | 2026-03-23 06:30:00.000000+0000 | КПП-Восточный |    12349 | Ремонтная служба |  Андрей Новиков |                            null |       True |
  работа | flexible
        1001 | 2026-03-23 | 2026-03-23 04:45:00.000000+0000 |  КПП-Северный |    12345 |  Оперативный цех |     Иван Петров |                            null |       True |
  работа |      day
        1001 | 2026-03-22 | 2026-03-22 04:30:00.000000+0000 |  КПП-Северный |    12345 |  Оперативный цех |     Иван Петров | 2026-03-22 16:30:00.000000+0000 |      False |
  работа |      day
        1003 | 2026-03-23 | 2026-03-23 16:45:00.000000+0000 |  КПП-Северный |    12347 |  Оперативный цех | Алексей Смирнов |                            null |       True |
  работа |    night
        1002 | 2026-03-23 | 2026-03-23 04:50:00.000000+0000 | КПП-Восточный |    12346 |  Цех переработки |   Сергей Иванов |                            null |       True |
  работа |      day
        1002 | 2026-03-22 | 2026-03-22 04:45:00.000000+0000 | КПП-Восточный |    12346 |  Цех переработки |   Сергей Иванов | 2026-03-22 16:45:00.000000+0000 |      False |
  работа |      day
```

```
SELECT * FROM presence_by_shift;
```

```
cqlsh:refinery_personnel> SELECT * FROM presence_by_shift;

 shift    | employee_id | access_gate    | department       | employee_name   | entry_time
----------+-------------+----------------+------------------+-----------------+---------------------------------
 flexible |        1005 | КПП-Восточный | Ремонтная служба |  Андрей Новиков | 2026-03-23 06:30:00.000000+0000
    night |        1003 |  КПП-Северный |  Оперативный цех | Алексей Смирнов | 2026-03-23 16:45:00.000000+0000
      day |        1001 |  КПП-Северный |  Оперативный цех |     Иван Петров | 2026-03-23 04:45:00.000000+0000
      day |        1002 | КПП-Восточный |  Цех переработки |   Сергей Иванов | 2026-03-23 04:50:00.000000+0000
      day |        1004 |  КПП-Западный |      Лаборатория |   Михаил Волков | 2026-03-23 05:15:00.000000+0000
```

```
SELECT * FROM access_log;
```

```
cqlsh:refinery_personnel> SELECT * FROM access_log;

 access_gate    | access_date | access_time                     | action | badge_id | employee_id | employee_name
----------------+-------------+---------------------------------+--------+----------+-------------+-----------------
  КПП-Северный |  2026-03-23 | 2026-03-23 16:45:00.000000+0000 |  entry |    12347 |        1003 | Алексей Смирнов
  КПП-Северный |  2026-03-23 | 2026-03-23 04:45:00.000000+0000 |  entry |    12345 |        1001 |    Иван Петров
 КПП-Восточный |  2026-03-23 | 2026-03-23 06:30:00.000000+0000 |  entry |    12349 |        1005 |  Андрей Новиков
 КПП-Восточный |  2026-03-23 | 2026-03-23 04:50:00.000000+0000 |  entry |    12346 |        1002 |  Сергей Иванов
  КПП-Западный |  2026-03-23 | 2026-03-23 05:15:00.000000+0000 |  entry |    12348 |        1004 |  Михаил Волков
```

```
#демонстрация выбора данных из таблиц
SELECT shift, COUNT(*) as employees_count
FROM presence_by_shift
GROUP BY shift;
```

```
 shift    | employees_count
----------+-----------------
 flexible |               1
    night |               1
      day |               3
```

```
SELECT employee_id, employee_name, department, entry_time, shift, access_gate
FROM current_presence;
```

employee_id	employee_name	department	entry_time	shift	access_gate
1004	Михаил Волков	Лаборатория	2026-03-23 05:15:00.000000+0000	day	КПП-Западный
1005	Андрей Новиков	Ремонтная служба	2026-03-23 06:30:00.000000+0000	flexible	КПП-Восточный
1001	Иван Петров	Оперативный цех	2026-03-23 04:45:00.000000+0000	day	КПП-Северный
1003	Алексей Смирнов	Оперативный цех	2026-03-23 16:45:00.000000+0000	night	КПП-Северный
1002	Сергей Иванов	Цех переработки	2026-03-23 04:50:00.000000+0000	day	КПП-Восточный

```
SELECT entry_date, entry_time, exit_time, department, shift, access_gate
FROM personnel_entries
WHERE employee_id = 1001
AND entry_date >= '2026-03-20'
AND entry_date <= '2026-03-23';
```

entry_date	entry_time	exit_time	department	shift	access_gate
2026-03-23	2026-03-23 04:45:00.000000+0000	null	Оперативный цех	day	КПП-Северный
2026-03-22	2026-03-22 04:30:00.000000+0000	2026-03-22 16:30:00.000000+0000	Оперативный цех	day	КПП-Северный

```
SELECT access_time, employee_id, employee_name, action
FROM access_log
WHERE access_gate = 'КПП-Северный'
AND access_date = '2026-03-23';
```

access_time	employee_id	employee_name	action
2026-03-23 16:45:00.000000+0000	1003	Алексей Смирнов	entry
2026-03-23 04:45:00.000000+0000	1001	Иван Петров	entry

```
SELECT employee_id, employee_name, entry_time, shift,
       (toUnixTimestamp(now()) - entry_timestamp) / 3600 AS hours_on_site
FROM current_presence
ALLOW FILTERING;
```

employee_id	employee_name	entry_time	shift	hours_on_site
1004	Михаил Волков	2026-03-23 05:15:00.000000+0000	day	492448593
1005	Андрей Новиков	2026-03-23 06:30:00.000000+0000	flexible	492448592
1001	Иван Петров	2026-03-23 04:45:00.000000+0000	day	492448594
1003	Алексей Смирнов	2026-03-23 16:45:00.000000+0000	night	492448583
1002	Сергей Иванов	2026-03-23 04:50:00.000000+0000	day	492448594

```
#обновляем данные в таблицах
UPDATE personnel_entries
SET exit_time = '2026-03-23 19:30:00',
    is_present = false
WHERE employee_id = 1001
AND entry_date = '2026-03-23'
AND entry_time = '2026-03-23 07:45:00';

#удаляем данные из таблиц
DELETE FROM current_presence
WHERE employee_id = 1001;

DELETE FROM presence_by_shift
WHERE shift = 'day'
AND employee_id = 1001;


#вписываем новые данные
INSERT INTO access_log (access_gate, access_date, access_time, employee_id, employee_name, action, badge_id)
VALUES ('КПП-Северный', '2026-03-23', '2026-03-23 19:30:00', 1001, 'Иван Петров', 'exit', 12345);

SELECT * FROM personnel_entries;
```

```
cqlsh:refinery_personnel> SELECT * FROM personnel_entries;

 employee_id | entry_date | entry_time                      | access_gate   | badge_id | department       | employee_name   | exit_time                       | is_present |
 purpose | shift
-------------+------------+---------------------------------+---------------+----------+------------------+-----------------+---------------------------------+------------+
---------+---------
        1004 | 2026-03-23 | 2026-03-23 05:15:00.000000+0000 | КПП-Западный |    12348 |      Лаборатория |   Михаил Волков |                            null |       True |
  работа |      day
        1005 | 2026-03-23 | 2026-03-23 06:30:00.000000+0000 | КПП-Восточный |    12349 | Ремонтная служба |   Андрей Новиков |                            null |       True |
  работа | flexible
        1001 | 2026-03-23 | 2026-03-23 04:45:00.000000+0000 |  КПП-Северный |    12345 |  Оперативный цех |      Иван Петров | 2026-03-23 16:30:00.000000+0000 |      False |
  работа |      day
        1001 | 2026-03-22 | 2026-03-22 04:30:00.000000+0000 |  КПП-Северный |    12345 |  Оперативный цех |      Иван Петров | 2026-03-22 16:30:00.000000+0000 |      False |
  работа |      day
        1003 | 2026-03-23 | 2026-03-23 16:45:00.000000+0000 |  КПП-Северный |    12347 |  Оперативный цех | Алексей Смирнов |                            null |       True |
  работа |    night
        1002 | 2026-03-23 | 2026-03-23 04:50:00.000000+0000 | КПП-Восточный |    12346 |  Цех переработки |   Сергей Иванов |                            null |       True |
  работа |      day
        1002 | 2026-03-22 | 2026-03-22 04:45:00.000000+0000 | КПП-Восточный |    12346 |  Цех переработки |   Сергей Иванов | 2026-03-22 16:45:00.000000+0000 |      False |
  работа |      day
```

```
SELECT shift, COUNT(*) as employees_count
FROM presence_by_shift
GROUP BY shift;
```

```
 shift     | employees_count
-----------+-----------------
 flexible |               1
    night |               1
      day |               2
```

```
SELECT employee_id, employee_name, entry_time, shift,
       (toUnixTimestamp(now()) - entry_timestamp) / 3600 as hours_on_site
FROM current_presence
ALLOW FILTERING;
```

```
 employee_id | employee_name    | entry_time                      | shift     | hours_on_site
-------------+------------------+---------------------------------+-----------+----------------
        1004 |   Михаил Волков | 2026-03-23 05:15:00.000000+0000 |       day |      492448662
        1005 |  Андрей Новиков | 2026-03-23 06:30:00.000000+0000 |  flexible |      492448661
        1003 | Алексей Смирнов | 2026-03-23 16:45:00.000000+0000 |     night |      492448651
        1002 |   Сергей Иванов | 2026-03-23 04:50:00.000000+0000 |       day |      492448662
```

2.1.2 Задача: Мониторинг параметров

Решение задачи «Мониторинг параметров» позволяет отслеживать и уведомлять о выходе параметров за установленные границы.

Все приведённые команды выполняются в терминале утилиты cqlsh:

```
# Создание таблицы с TTL 24 часа
CREATE TABLE iot_sensors_data (
    sensor_id INT,                    -- ID датчика (1001-9999)
    home_id INT,                      -- ID дома (1-1000)
    timestamp TIMESTAMP,
    temperature DECIMAL,              -- температура в °C
    humidity INT,                     -- влажность в %
    co2_level INT,                    -- уровень CO2 в ppm
    is_alert BOOLEAN,                 -- флаг аварии
    battery_level INT,                -- заряд батареи в %
    PRIMARY KEY ((sensor_id), timestamp)
) WITH default_time_to_live = 86400;

#Создание индекса для поиска аварий
CREATE INDEX ON iot_sensors_data (is_alert);
CREATE INDEX ON iot_sensors_data (home_id);

# Заполнение реальными данными

# Датчик №1001 (Гостиная, дом №42): нормальные показатели
INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
```

```
VALUES (1001, 42, '2026-03-23 08:00:00', 22.5, 45, 410, false, 94);

INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (1001, 42, '2026-03-23 12:00:00', 23.2, 42, 520, false, 94);

INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (1001, 42, '2026-03-23 18:00:00', 24.1, 40, 480, false, 93);

# Датчик №2047 (Котельная, дом №156): высокая температура (авария)
INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (2047, 156, '2026-03-23 09:30:00', 42.8, 18, 890, true, 67);

# Датчик №3089 (Улица, дом №78): низкая температура (зимний режим)
INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (3089, 78, '2026-03-23 06:00:00', -8.5, 88, 395, true, 52);

# Датчик №4523 (Детская, дом №201): высокий CO2 (нет проветривания)
INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (4523, 201, '2026-03-23 14:20:00', 23.8, 55, 1850, true, 88);

#Датчик №5098 (Кухня, дом №33): разряженная батарея
INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (5098, 33, '2026-03-23 11:00:00', 21.3, 52, 440, false, 12);

# Тестовые данные с коротким TTL (через 30 секунд удалятся)
INSERT INTO iot_sensors_data (sensor_id, home_id, timestamp, temperature, humidity, co2_level, is_alert,
battery_level)
VALUES (9999, 999, '2026-03-23 15:00:00', 20.0, 50, 400, false, 100)
USING TTL 30;

# Демонстрация возможностей

# Найти все аварийные датчики прямо сейчас:
SELECT sensor_id, home_id, timestamp, temperature, co2_level, battery_level
FROM iot_sensors_data
WHERE is_alert = true
ALLOW FILTERING;
```

```
 sensor_id | home_id | timestamp                       | temperature | co2_level | battery_
level
-----------+---------+---------------------------------+-------------+-----------+---------
------
      3089 |      78 | 2026-03-23 03:00:00.000000+0000 |        -8.5 |       395 |
   52
      4523 |     201 | 2026-03-23 11:20:00.000000+0000 |        23.8 |      1850 |
   88
      2047 |     156 | 2026-03-23 06:30:00.000000+0000 |        42.8 |       890 |
   67
```

```
# Получить суточный график температуры для датчика №1001:
SELECT timestamp, temperature, humidity
FROM iot_sensors_data
WHERE sensor_id = 1001
AND timestamp >= '2026-03-23'
AND timestamp < '2026-03-24';
```

```
timestamp                       | temperature | humidity
--------------------------------+-------------+----------
2026-03-23 05:00:00.000000+0000 |        22.5 |       45
2026-03-23 09:00:00.000000+0000 |        23.2 |       42
2026-03-23 15:00:00.000000+0000 |        24.1 |       40
```

2.1.3 Задача: Учет материалов на складе производства труб

Решение задачи «Учет материалов на складе производства труб» позволяет записывать в БД данные о наличии на складе продукции, о ее количестве и о ее статусе (брони).

Все приведённые команды выполняются в терминале утилиты cqlsh:

```
#создание базы данных
CREATE KEYSPACE IF NOT EXISTS pipeline_warehouse
WITH replication = {
  'class': 'NetworkTopologyStrategy',
  'datacenter1': 3
};

#подключение к БД
USE pipeline_warehouse;

#создание таблиц
CREATE TABLE IF NOT EXISTS inventory (
    product_id INT,
    warehouse_code TEXT,
    quantity COUNTER,
    PRIMARY KEY ((product_id, warehouse_code))
);

CREATE TABLE IF NOT EXISTS inventory_movements (
    movement_id INT,
    product_id INT,
    warehouse_code TEXT,
    movement_type TEXT,
    quantity INT,
    order_reference TEXT,
    created_at TIMESTAMP,
    operator TEXT,
    PRIMARY KEY ((product_id, warehouse_code), created_at, movement_id)
) WITH CLUSTERING ORDER BY (created_at DESC);

CREATE TABLE IF NOT EXISTS orders (
    order_id INT,
    customer_code INT,
    order_date DATE,
    status TEXT,
    delivery_address TEXT,
    total_weight DECIMAL,
    total_amount DECIMAL,
    products MAP<INT, INT>,
    created_at TIMESTAMP,
    updated_at TIMESTAMP,
    PRIMARY KEY ((customer_code), order_date, order_id)
) WITH CLUSTERING ORDER BY (order_date DESC);

CREATE TABLE IF NOT EXISTS products_catalog (
    product_id INT PRIMARY KEY,
    sku TEXT,
    name TEXT,
    category TEXT,
    standard TEXT,
    material TEXT,
    diameter DECIMAL,
    wall_thickness DECIMAL,
    pressure_rating TEXT,
    attributes MAP<TEXT, TEXT>,
    unit_of_measure TEXT,
    weight_per_unit DECIMAL,
    base_price DECIMAL,
    min_stock_quantity INT,
    is_active BOOLEAN
```

```
);

#заполнение таблиц данными
INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, wall_thickness, pressure_rating, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (1001, 'PIPE-219-6-20', 'Труба 219х6 ст.20', 'pipe', 'ГОСТ 10704-91', 'Сталь 20', 219.00, 6.00, '16', 'm', 31.52, 1850.00, 500, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, wall_thickness, pressure_rating, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (1002, 'PIPE-159-45-12H18N10T', 'Труба 159х4.5 12Х18Н10Т', 'pipe', 'ГОСТ 9941-81', '12Х18Н10Т', 159.00, 4.50, '25', 'm', 17.15, 4250.00, 300, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, wall_thickness, pressure_rating, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (1003, 'PIPE-325-8-09G2S', 'Труба 325х8 09Г2С', 'pipe', 'ГОСТ 10704-91', '09Г2С', 325.00, 8.00, '25', 'm', 62.54, 3250.00, 400, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, attributes, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (2001, 'ELBOW-90-219-20', 'Отвод 90° 219х6 ст.20', 'fitting', 'ГОСТ 17375-2001', 'Сталь 20', 219.00, {'type': 'крутоизогнутый', 'radius': '1.5D', 'coating': 'грунтовка'}, 'pcs', 45.20, 5200.00, 150, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, attributes, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (2002, 'TEE-159-159-12H18N10T', 'Тройник 159х159 12Х18Н10Т', 'fitting', 'ГОСТ 17376-2001', '12Х18Н10Т', 159.00, {'type': 'равнопроходный', 'connection': 'сварной'}, 'pcs', 28.30, 8750.00, 80, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, pressure_rating, attributes, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (3001, 'GATE-150-25-20', 'Задвижка клиновая DN150 Ру25', 'valve', 'ГОСТ 5762-2002', 'Сталь 20', 150.00, '25', {'actuator': 'ручной', 'flange_type': 'плоский', 'seal': 'резина'}, 'pcs', 185.00, 28500.00, 50, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, pressure_rating, attributes, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (3002, 'BALL-100-40-12H18N10T', 'Кран шаровой DN100 Ру40', 'valve', 'ГОСТ Р 52760-2007', '12Х18Н10Т', 100.00, '40', {'actuator': 'рукоятка', 'body': 'нержавейка', 'ball': 'полированный'}, 'pcs', 42.50, 18750.00, 120, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, diameter, attributes, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (4001, 'FLANGE-219-16-20', 'Фланец плоский DN200 Ру16', 'flange', 'ГОСТ 12820-80', 'Сталь 20', 200.00, {'type': 'плоский', 'sealing': 'выступ-впадина'}, 'pcs', 12.80, 2850.00, 200, true);

INSERT INTO products_catalog (product_id, sku, name, category, standard, material, attributes, unit_of_measure, weight_per_unit, base_price, min_stock_quantity, is_active)
VALUES (5001, 'BOLT-M20-100-8.8', 'Болт М20х100 оцинк. 8.8', 'fastener', 'ГОСТ 7798-70', 'Сталь 40Х', {'coating': 'цинк', 'strength': '8.8', 'thread': 'полная'}, 'set', 0.32, 45.00, 5000, true);

#обновление данных
UPDATE inventory SET quantity = quantity + 1450 WHERE product_id = 1001 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 890 WHERE product_id = 1002 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 2340 WHERE product_id = 1003 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 520 WHERE product_id = 2001 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 310 WHERE product_id = 2002 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 185 WHERE product_id = 3001 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 120 WHERE product_id = 3002 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 450 WHERE product_id = 4001 AND warehouse_code = 'MAIN';
UPDATE inventory SET quantity = quantity + 5000 WHERE product_id = 5001 AND warehouse_code = 'MAIN';

UPDATE inventory SET quantity = quantity + 500 WHERE product_id = 1001 AND warehouse_code = 'RESERVE';
UPDATE inventory SET quantity = quantity + 200 WHERE product_id = 1002 AND warehouse_code = 'RESERVE';
UPDATE inventory SET quantity = quantity + 50 WHERE product_id = 3001 AND warehouse_code = 'RESERVE';

BEGIN UNLOGGED BATCH
    UPDATE inventory SET quantity = quantity - 50 WHERE product_id = 1001 AND warehouse_code = 'MAIN';
    UPDATE inventory SET quantity = quantity - 10 WHERE product_id = 2001 AND warehouse_code = 'MAIN';
APPLY BATCH;

#заполнение данными
```

```
        INSERT INTO inventory_movements (movement_id, product_id, warehouse_code, movement_type, quantity,
order_reference, created_at, operator)
        VALUES (1, 1001, 'MAIN', 'outgoing', 50, 'ORD-20260323-001', toTimestamp(now()), 'operator1');

        INSERT INTO inventory_movements (movement_id, product_id, warehouse_code, movement_type, quantity,
order_reference, created_at, operator)
        VALUES (2, 2001, 'MAIN', 'outgoing', 10, 'ORD-20260323-001', toTimestamp(now()), 'operator1');

        INSERT INTO orders (order_id, customer_code, order_date, status, delivery_address, total_weight,
total_amount, products, created_at, updated_at)
        VALUES (10001, 1001, '2026-03-23', 'shipped', 'г. Тюмень, ул. Нефтяников, 15',
                2028.00,
                144500.00,
                {1001: 50, 2001: 10},
                toTimestamp(now()), toTimestamp(now()));

        #обновление данных
        BEGIN UNLOGGED BATCH
            UPDATE inventory SET quantity = quantity - 2 WHERE product_id = 3001 AND warehouse_code = 'MAIN';
            UPDATE inventory SET quantity = quantity - 8 WHERE product_id = 4001 AND warehouse_code = 'MAIN';
            UPDATE inventory SET quantity = quantity - 32 WHERE product_id = 5001 AND warehouse_code = 'MAIN';
        APPLY BATCH;

        #внесение новых данных
        INSERT INTO inventory_movements (movement_id, product_id, warehouse_code, movement_type, quantity,
order_reference, created_at, operator)
        VALUES (3, 3001, 'MAIN', 'outgoing', 2, 'ORD-20260323-002', toTimestamp(now()), 'operator2');

        INSERT INTO inventory_movements (movement_id, product_id, warehouse_code, movement_type, quantity,
order_reference, created_at, operator)
        VALUES (4, 4001, 'MAIN', 'outgoing', 8, 'ORD-20260323-002', toTimestamp(now()), 'operator2');

        INSERT INTO inventory_movements (movement_id, product_id, warehouse_code, movement_type, quantity,
order_reference, created_at, operator)
        VALUES (5, 5001, 'MAIN', 'outgoing', 32, 'ORD-20260323-002', toTimestamp(now()), 'operator2');

        INSERT INTO orders (order_id, customer_code, order_date, status, delivery_address, total_weight,
total_amount, products, created_at, updated_at)
        VALUES (10002, 1002, '2026-03-23', 'confirmed', 'г. Казань, ул. Химиков, 8',
                482.64,
                81240.00,
                {3001: 2, 4001: 8, 5001: 32},
                toTimestamp(now()), toTimestamp(now()));

        BEGIN UNLOGGED BATCH
            UPDATE inventory SET quantity = quantity + 10 WHERE product_id = 2001 AND warehouse_code = 'MAIN';
        APPLY BATCH;

        INSERT INTO inventory_movements (movement_id, product_id, warehouse_code, movement_type, quantity,
order_reference, created_at, operator)
        VALUES (6, 2001, 'MAIN', 'incoming', 10, 'ORD-20260323-003-CANCEL', toTimestamp(now()), 'operator1');

        INSERT INTO orders (order_id, customer_code, order_date, status, delivery_address, total_weight,
total_amount, products, created_at, updated_at)
        VALUES (10003, 1001, '2026-03-23', 'cancelled', 'г. Тюмень, ул. Нефтяников, 15',
                0, 0, {2001: 10},
                toTimestamp(now()), toTimestamp(now()));

        #демонстрация таблиц
        SELECT * FROM inventory;
```

```
 product_id | warehouse_code | quantity
------------+----------------+----------
       1002 |           MAIN |      890
       1003 |           MAIN |     2340
       1001 |           MAIN |     1400
       1002 |        RESERVE |      200
       1001 |        RESERVE |      500
       5001 |           MAIN |     4968
       2001 |           MAIN |      520
       3002 |           MAIN |      120
       3001 |        RESERVE |       50
       4001 |           MAIN |      442
       3001 |           MAIN |      183
       2002 |           MAIN |      310
```

```
SELECT * FROM inventory_movements;
```

```
cqlsh:pipeline_warehouse> SELECT * FROM inventory_movements;

 product_id | warehouse_code | created_at                      | movement_id | movement_ty
pe | operator  | order_reference          | quantity
------------+----------------+---------------------------------+-------------+-----------
---+-----------+--------------------------+----------
       1001 |           MAIN | 2026-03-26 22:40:37.000000+0000 |           1 |      outgoi
ng | operator1 |        ORD-20260323-001 |       50
       5001 |           MAIN | 2026-03-26 22:41:09.906000+0000 |           5 |      outgoi
ng | operator2 |        ORD-20260323-002 |       32
       2001 |           MAIN | 2026-03-26 22:41:18.874000+0000 |           6 |      incomi
ng | operator1 | ORD-20260323-003-CANCEL |       10
       2001 |           MAIN | 2026-03-26 22:40:37.965000+0000 |           2 |      outgoi
ng | operator1 |        ORD-20260323-001 |       10
       4001 |           MAIN | 2026-03-26 22:41:09.902000+0000 |           4 |      outgoi
ng | operator2 |        ORD-20260323-002 |        8
       3001 |           MAIN | 2026-03-26 22:41:09.898000+0000 |           3 |      outgoi
ng | operator2 |        ORD-20260323-002 |        2
```

```
SELECT * FROM orders;
```

```
cqlsh:pipeline_warehouse> SELECT * FROM orders;

 customer_code | order_date | order_id | created_at                      | delivery_addres
s              | products                    | status    | total_amount | total_weight |
updated_at
---------------+------------+----------+---------------------------------+----------------
---------------+-----------------------------+-----------+--------------+--------------+-
--------------------------------
          1001 | 2026-03-23 |    10001 | 2026-03-26 22:40:47.355000+0000 | г. Тюмень, ул.
Нефтяников, 15 |         {1001: 50, 2001: 10} |   shipped |    144500.00 |      2028.00 |
2026-03-26 22:40:47.355000+0000
          1001 | 2026-03-23 |    10003 | 2026-03-26 22:41:19.885000+0000 | г. Тюмень, ул.
Нефтяников, 15 |                  {2001: 10} | cancelled |            0 |            0 |
2026-03-26 22:41:19.885000+0000
          1002 | 2026-03-23 |    10002 | 2026-03-26 22:41:11.100000+0000 |     г. Казань,
ул. Химиков, 8 | {3001: 2, 4001: 8, 5001: 32} | confirmed |     81240.00 |       482.64 |
2026-03-26 22:41:11.100000+0000
```

```
SELECT * FROM products_catalog;
```

```
cqlsh:pipeline_warehouse> SELECT * FROM products_catalog;

 product_id | attributes                                                
```

```
 order_id | customer_code | order_date | status    | total_weight | total_amount | price_
per_kg
----------+---------------+------------+-----------+--------------+--------------+-------
--------------------------------
    10001 |          1001 | 2026-03-23 |   shipped |      2028.00 |    144500.00 |  71.25
246548323471400394477317554241
    10002 |          1002 | 2026-03-23 | confirmed |       482.64 |     81240.00 | 168.32
421680755842864246643460964694
```

```
SELECT sku, name, base_price
FROM products_catalog
WHERE is_active = true
ALLOW FILTERING;
```

```
 timestamp                        | temperature | humidity
----------------------------------+-------------+----------
 2026-03-23 05:00:00.000000+0000 |        22.5 |       45
 2026-03-23 09:00:00.000000+0000 |        23.2 |       42
 2026-03-23 15:00:00.000000+0000 |        24.1 |       40
```

```
 sku                  | name                          | base_price
----------------------+-------------------------------+------------
 BALL-100-40-12H18N10T |      Кран шаровой DN100 Ру40 |   18750.00
      BOLT-M20-100-8.8 |      Болт М20х100 оцинк. 8.8 |      45.00
        GATE-150-25-20 | Задвижка клиновая DN150 Ру25 |   28500.00
         PIPE-219-6-20 |            Труба 219х6 ст.20 |    1850.00
       ELBOW-90-219-20 |       Отвод 90° 219х6 ст.20 |    5200.00
 TEE-159-159-12H18N10T |    Тройник 159х159 12Х18Н10Т |    8750.00
      FLANGE-219-16-20 |    Фланец плоский DN200 Ру16 |    2850.00
      PIPE-325-8-09G2S |            Труба 325х8 09Г2С |    3250.00
 PIPE-159-45-12H18N10T |      Труба 159х4.5 12Х18Н10Т |    4250.00
```

2.2 Моделирование систем распределённой архитектуры

2.2.1 Задача: Создание однокластерной системы с двумя узлами данных

Решение задачи «Создание однокластерной системы с двумя узлами данных» позволяет создать модель данных, имеющую один кластер, в котором располагаются два узла данных.

Все настройки производятся в файле database.yaml, который находится в директории ~/conf/database/ (~ путь, по которому лежит программа)

```
# НАСТРОЙКА УЗЛА ДАННЫХ 1

cluster_name: 'TwoNodeCluster'
# Уникальное имя для всего кластера. Должно быть одинаковым на всех узлах.

# ---- Настройка сетевого взаимодействия ----
# IP-адрес, который узел будет использовать для внутреннего общения с другими узлами кластера.
listen_address: 192.168.1.101

# IP-адрес или доменное имя, которое будут использовать клиентские приложения (например, cqlsh).
rpc_address: 192.168.1.101

# ---- Настройка сидов (Seeds) ----
# Список узлов, которые будут служить «точками входа» для новых узлов при присоединении к кластеру.
# Для кластера из двух узлов можно указать оба.
seed_provider:
```

```
    - class_name: org.quadra.database.locator.SimpleSeedProvider
      parameters:
          - seeds: "192.168.1.101,192.168.1.102"

# ---- Настройка для нового узла ----
# При создании первого узла в новом кластере эта опция должна быть false.
auto_bootstrap: false


# НАСТРОЙКА УЗЛА ДАННЫХ 2

# То же самое имя кластера
cluster_name: 'TwoNodeCluster'

# ---- Сетевые настройки Узла 2 ----
listen_address: 192.168.1.102
rpc_address: 192.168.1.102

# ---- Сиды (Seeds) - список ТОТ ЖЕ САМЫЙ, что и на Узле 1 ----
seed_provider:
    - class_name: org.quadra.database.locator.SimpleSeedProvider
      parameters:
          - seeds: "192.168.1.101,192.168.1.102"

# ---- Настройка для нового узла ----
# Для всех узлов, которые присоединяются к уже работающему кластеру, эта опция должна быть true.
# Она позволяет узлу автоматически перенести (stream) необходимые данные.
auto_bootstrap: true


# ЗАПУСК И ПРОВЕРКА СТАТУСА ДВУХ УЗЛОВ
nodetool status


# СОЗДАНИЕ БАЗЫ ДАННЫХ
# Подключение к любому узлу с помощью утилиты cqlsh
cqlsh 192.168.1.101 9042 -u quadradb -p quadradb

# Создание БД со стратегией репликации SimpleStrategy и фактором репликации 2

CREATE KEYSPACE IF NOT EXISTS my_two_node_app
WITH REPLICATION = {
    'class' : 'SimpleStrategy',
    'replication_factor' : 2
};

# Переключение на новую базу данных и создание таблицы
USE my_two_node_app;

CREATE TABLE users (
    user_id UUID PRIMARY KEY,
    name TEXT,
    email TEXT
);
```

2.2.2 *Задача: Добавление узла в кластер*

Решение задачи «Добавление узла в кластер» позволяет добавить новый узел в созданную ранее однокластерную систему с двумя узлами.

Перед добавлением нового узла в кластер необходимо:

1. Установите на новый сервер ту же версию СУБД «Квадра» и OpenJDK.
2. Убедитесь, что на новом узле настроена синхронизация времени (NTP) и открыты сетевые порты для общения с кластером.

3. Настроить файл конфигурации database.yaml на новом узле и отредактируйте следующие параметры:

```
# Конфигурации для нового узла (10.0.0.4)
cluster_name: 'TwoNodeCluster'
seed_provider:
  - class_name: org.quadra.database.locator.SimpleSeedProvider
    parameters:
      - seeds: "10.0.0.1,10.0.0.2"  # IP существующих seed-узлов
listen_address: 10.0.0.4             # IP нового узла
rpc_address: 10.0.0.4
auto_bootstrap: true                 # Включает копирование данных

# Важно: Список seeds должен быть одинаковым на всех узлах кластера
```

4. Запустите службу database на новом узле. В процессе запуска узел свяжется с seed-узлами, присоединится к кольцу и начнет копировать данные.

3 Контакты

Для контактов с командой разработчиков просьба обращаться по электронной почте: info@quadra-system.ru.