## ПРАКТИЧЕСКАЯ РАБОТА

***Тема: Нормализация реляционной модели данных методом декомпозиции отношений***

**Порядок выполнения работы**

1. Для указанного варианта задания выполнить нормализацию отношений методом декомпозиции с учетом выделенных функциональных зависимостей.
2. Представить результат в виде *нормализованной реляционной модели*.
3. Оформить отчет по работе.

**Содержание отчета**

1. Титульный лист.
2. Реляционная таблица для заданного варианта.
3. Описание первичных ключей и функциональных зависимостей для заданного отношения (таблицы).
4. Нормализованная реляционная модель до уровня 3НФ.

## Теоретические сведения

*Нормализация* – метод создания набора отношений с заданными свойствами на основе некоторых требований к данным. Процесс нормализации – формальный метод для оптимизации столбцов отношений и устранения аномалий.

**Избыточность данных и аномалии обновления**

Основная цель проектирования реляционной БД – группирование атрибутов в отношениях таким образом, чтобы минимизировать избыточность данных (сокращение объема вторичной памяти для хранения БД) и повышение надежности при работе с данными.

Обычно процесс проектирования отношений реляционной БД ведется на основе разработанной ER-диаграммы или на основе просто здравого смысла разработчика. В общем случае при таком подходе расположение атрибутов в отношениях *неоптимальное*. При работе с отношениями, содержащими избыточные данные, могут возникать проблемы – *аномалии обновления*.

Аномалии обновления делят на три вида:

* + - * *аномалии вставки* – возникают при добавлении новых несогласованных данных (нарушающих целостность данных в отношении);
      * *аномалии изменения* – возникают при изменении части ранее введенных данных; частичное обновление сведений приведет к нарушению целостности данных отношения;
      * *аномалии удаления* – возникают при удалении строк из отношений.

Обычно для решения проблем избыточности и аномалий выполняется деление отношения на такие отношения, в которых избыточности не будет. Для выполнения такого процесса необходимо выявить все зависимости между атрибутами отношения (потеря одной такой зависимости меняет модель внешнего мира).

**Функциональные зависимости**

Выявление смысловой зависимости между данными – один из способов формализации смысловой информации о данных.

*Функциональная зависимость* описывает связь типа «многие-к-одному» между атрибутами отношения, где «много» – детерминант функциональной зависимости. Функциональная зависимость является семантическим свойством атрибутов отношения.

Если в отношении *R*, содержащем атрибуты *A* и *B*, атрибут *B* функционально зависит от атрибута *A* (*А* является детерминантом атрибута *B*) *A*  *B* , то в каждом кортеже этого отношения каждое конкретное значение атрибута *A* всегда связано только с одним значением атрибута *B*.

Особенности функциональных зависимостей, лежащие в основе процесса нормализации:

* + - * функциональная зависимость является специализированным правилом целостности – она накладывает ограничения на допустимые значения атрибутов отношений; эту особенность можно использовать при обновлении БД, т.к. зная, какие функциональные зависимости есть в отношении, можно понять, нарушат ли новые данные целостность данных отношения;
      * функциональная зависимость является обобщением понятия потенциального ключа; функциональные зависимости позволяют определить все потенциальные ключи отношения (и соответственно – первичный ключ): все атрибуты отношения, которые не являются частью первичного (или потенциального) ключа, должны функционально зависеть от этого ключа; если не все остальные атрибуты отношения зависят от некоторого детерминанта, то этот детерминант не является потенциальным ключом этого отношения.

**Нормальные формы и нормализация методом декомпозиции**

*Нормализация* – это формальный метод анализа отношений на основе их первичного ключа и существующих функциональных зависимостей.

Суть процесса нормализации:

* + - * в нормализованных отношениях не разрешаются никакие функциональные зависимости, кроме функциональных зависимостей вида *K*  *A* , где *K* – потенциальный ключ отношения *R*, а *A* – неключевой атрибут;
      * если же отношение *R* имеет функциональные зависимости *B*  *A* , где B не является потенциальным ключом, то в отношении *R* будет наблюдаться избыточность данных.

В процессе нормализации реляционных отношений применяются концепции *нормальных форм*. Говорят, что отношение находится в определенной нормальной форме, если оно удовлетворяет правилам этой нормальной формы. В настоящее время используется шесть нормальных форм, которые зависят друг от друга путем усложнения (вложенности) набора правил:

1*НФ*  2*НФ*  3*НФ*  *НФБК*  4*НФ*  5*НФ* .

Каждая нормальная форма, таким образом, **удовлетворяет всем предыдущим нормальным формам**. Более высокая нормальная форма приводит к более строгому формату отношения (меньшее число аномалий обновления).

**Примечание.** БД можно построить и на отношениях, находящихся в первой нормальной форме, но такая БД будет сильно подвержена аномалиям и избыточности данных.

**На практике желательно использовать, как минимум, *3НФ***, чтобы устранить большинство аномалий обновления.

1. *1НФ*. Отношение находится в *1НФ* тогда и только тогда, когда в любом допустимом значении этого отношения каждый кортеж содержит только одно значение для каждого из атрибутов, т.е. это значение не имеет внутренней структуры (множество, таблица и т.п.). Отношения в *1НФ* имеют большое количество аномалий обновления.
2. *2НФ*. Отношение находится в *2НФ* тогда и только тогда, когда оно находится в *1НФ*, и каждый атрибут отношения, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого первичного ключа.

**Полной функциональной зависимостью** называется такая зависимость *A*  *B* , когда *B* функционально зависит от *A* и не зависит ни от какого подмножества *A* (т.е. удаление какого-либо атрибута из *A* приведет к утрате этой функциональной зависимости). *2НФ* устраняет в отношении частичные функциональные зависимости неключевых атрибутов от первичного ключа, которые выносятся в отдельное отношение вместе с копиями своих детерминантов (частей первичного ключа, от которого они зависят).

1. *3НФ*. Отношение находится в *3НФ* тогда и только тогда, когда оно находится в *2НФ* и не имеет не входящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

Транзитивной функциональной зависимостью называется зависимость *A*  *C* , если существуют зависимости *A*  *B* и *B*  *C* (говорят, что атрибут *C* транзитивно зависит от *A* через атрибут *B*), при условии, что атрибут A функционально не зависит ни от атрибута B, ни от атрибута *C*.

*3НФ* устраняет в отношении транзитивные функциональные зависимости неключевых атрибутов от первичного ключа, которые выносятся в отдельное отношение вместе с копиями своих детерминантов. В *3НФ* устранено большинство аномалий от первичного ключа, но отношение в этой форме имеет аномалии в случае наличия более чем одного потенциального ключа.

*Декомпозиция* – формирование отношений БД путем разделения их на более мелкие, если эти отношения не выполняют правила необходимой нормальной формы.

## Рекомендации по выполнению работы

*Этап* 1. Выделить функциональные зависимости для каждого отношения исходной реляционной схемы. Проверить практический смысл выделенных функциональных зависимостей.

*Этап* 2. Для каждого отношения (включая и вновь создаваемые) последовательно применить правила нормальных форм. При несоблюдении текущего правила в отношении выполнить его декомпозицию (удалить проблемный атрибут из отношения с образованием нового отношения, первичным ключом которого будет детерминант рассматриваемой функциональной зависимости (этот атрибут только копируется в новое отношение)). Нормализованное отношение должно удовлетворять как минимум *3НФ*.

*Этап* 3. Для полученной нормализованной реляционной схемы проверить смысл ссылок.

*Этап* 4. Реализовать полученные реляционные отношения в виде таблиц в среде целевой СУБД.

*Этап* 5. Оформить отчет по работе.

**Пример приведения отношения к 3НФ**

Рассмотрим отношение «Экзаменационная ведомость»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код студента | Фамилия | Код экзамена | Предмет и дата | Оценка |
| 1 | Иванов | 1 | Математика, 05.06.2019 | 4 |
| 2 | Петров | 1 | Математика, 05.06.2019 | 5 |
| 1 | Иванов | 2 | Физика, 10.06.2019 | 5 |
| 2 | Петров | 2 | Физика, 10.06.2019 | 5 |

Первичный ключ таблицы состоит из атрибутов: Код студента, Код экзамена

**Отношение находится в первой нормальной форме (1НФ), если все атрибуты отношения принимают простые значения (атомарные или неделимые), не являющиеся множеством или кортежем из более элементарных составляющих.**

Наше отношение не находится в 1НФ.

Приведем отношение к 1НФ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код студента | Фамилия | Код экзамена | Предмет | Дата | Оценка |
| 1 | Иванов | 1 | Математика | 05.06.2019 | 4 |
| 2 | Петров | 1 | Математика | 05.06.2019 | 5 |
| 1 | Иванов | 2 | Физика | 10.06.2019 | 5 |
| 2 | Петров | 2 | Физика | 10.06.2019 | 5 |

Для исследования наличия 2НФ следует проанализировать функциональные зависимости между атрибутами отношения.

Единственный способ определить функциональные зависимости – внимательно проанализировать семантику (смысл) атрибутов.

Примеры функциональных зависимостей для отношения ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ:

Код студента → Фамилия

Код студента, Код экзамена → Оценка

Код экзамена → Дата

Код экзамена → Предмет

**Отношение находится в 2НФ, если оно находится в 1НФ и каждый неключевой атрибут зависит от всего первичного ключа (не зависит от части ключа).**

**Отношение находится в 3НФ, если оно находится в 2НФ и каждый ключевой атрибут нетранзитивно зависит от первичного ключа. Отношение находится в 3НФ в том и только том случае, если все неключевые атрибуты отношения взаимно независимы и полностью зависят от первичного ключа.**

Продолжим рассмотрение примера с отношением ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ. Для более краткой записи процесса нормализации введем следующие обозначения: КС – код студента, КЭ – код экзамена, Ф – фамилия, П – предмет, Д – дата, О – оценка.

Наше отношение примет вид: R=( **КС, КЭ**, Ф, П, Д, О )

Выпишем функциональные зависимости:

КС, КЭ → Ф, П, Д, О (КС, КЭ - первичный ключ отношения, все неключевые атрибуты зависят от первичного ключа)

При этом некоторые атрибуты зависят не от всего ключа в целом:

КЭ → П

КЭ → Д (предмет и дата зависят только от кода экзамена)

КС → Ф (фамилия студента зависит тольео от кода студента)

В соответствии с определением, **отношение находится во второй нормальной форме (2НФ), если оно находится в 1НФ и каждый неключевой атрибут зависит от первичного ключа и не зависит от части ключа**. Здесь атрибуты П, Д, Ф зависят от части ключа. Чтобы избавиться от этих зависимостей необходимо произвести декомпозицию отношения.

Выделим неполные зависимости в отдельные отношения. Если какие-то атрибуты зависят от одной части ключа, объединяем их в одну таблицу.

Получим отношение R1(**КС**, Ф) - это отношение находится в 2 НФ, так как ключ отношения простой и частичной зависимости быть не может. Так как в этом отношении нет транзитивных зависимостей, отношение R1(КС, Ф) находится в 3НФ.

Второе отношение R2(**КЭ**, П, Д) - зависимости неключевых атрибутов от части ключа нет, следовательно отношение находится в 2НФ. Транзитивных зависимостей в этом отношении также нет, следовательно отношение находится в 3НФ.

Исходное отношение приведено к виду: R(**КС, КЭ**, О). Из него выведены неключевые атрибуты, зависящие от части ключа. Неключевой атрибут О зависит от ключа КС, КЭ в целом, а не от его части. Значит, это отношение находится в 2НФ. Транзитивные зависимости отсутствуют, то есть отношение находится в 3НФ.

Таким образом все полученные отношения находятся в 3НФ.

Между таблицами установлены связи, как показано на рис. 1.

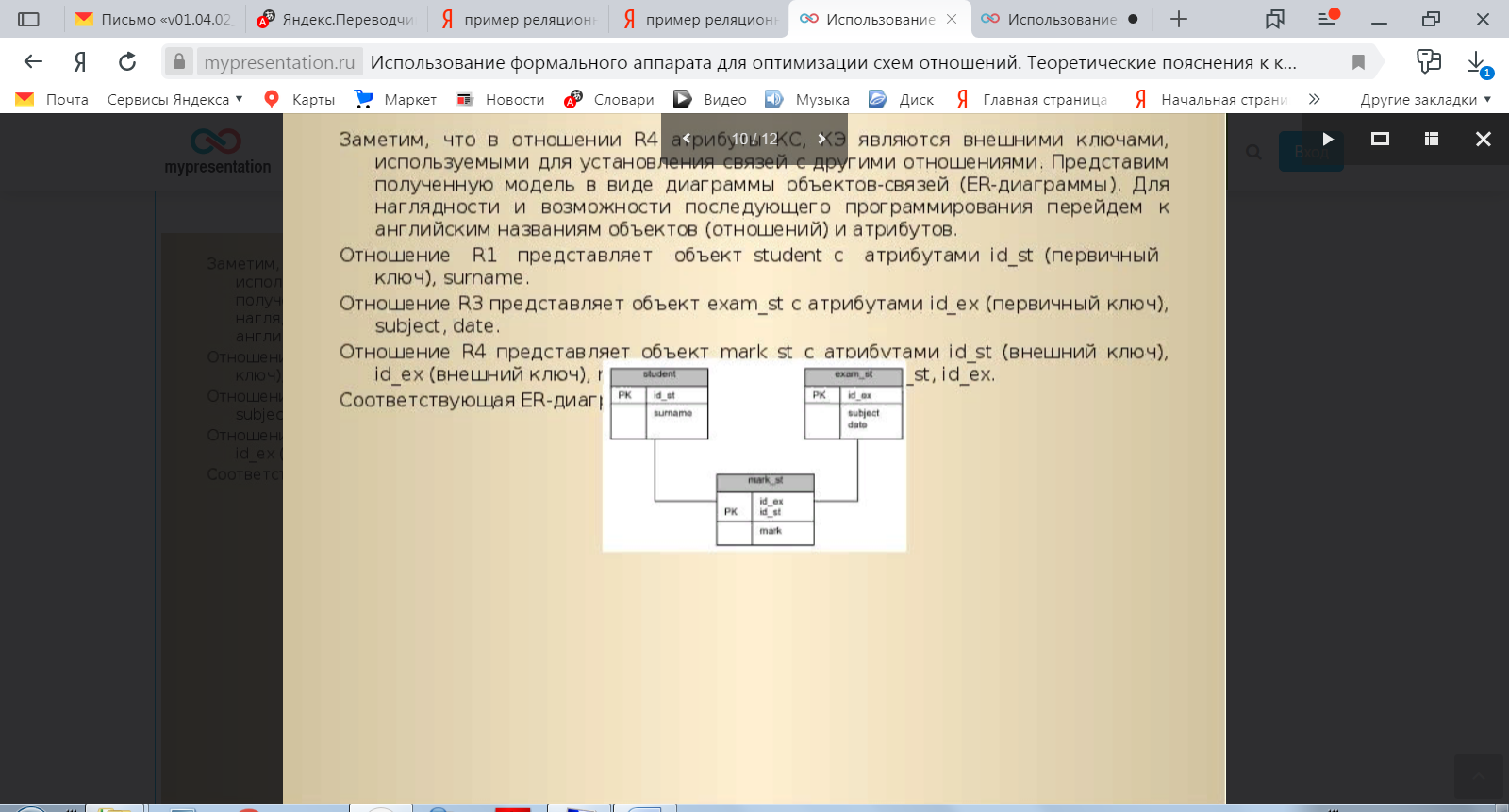


Рис. 1. Полученная реляционная модель (нормализованная)

Использованы следующие обозначения:

id\_st – код студента;

surname – фамилия;

id\_ex – код экзамена;

subject – предмет;

date – дата;

mark – оценка.

**Варианты заданий**

*Вариант 1 – отношение «Морские перевозки»*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер судна | Название | Номер рейса | Дата погрузки | Порт погрузки | Дата прибытия | Порт прибытия | Ф.И.О.  капитана | Вид судна | Грузо  подъем  ность, тонны |
| 526 | Japan Bear | 9201W | 5/31/92 | SFO | 6/6/92 | HNL | Емелин А.О. | Сухогруз | 500 |
| 603 | Korea Bear | 9202W | 5/05/92 | OAK | 6/19/92 | OSA | Крылов О.Б. | Ролкер | 1000 |
| 531 | China Bear | 9203W | 6/20/92 | LAX | 7/10/92 | PAP | Мухин Е.А. | Универсал | 1500 |
| 526 | Japan Bear | 9204W | 8/20/92 | SFO | 8/27/92 | HNL | Емелин А.О. | Сухогруз | 500 |

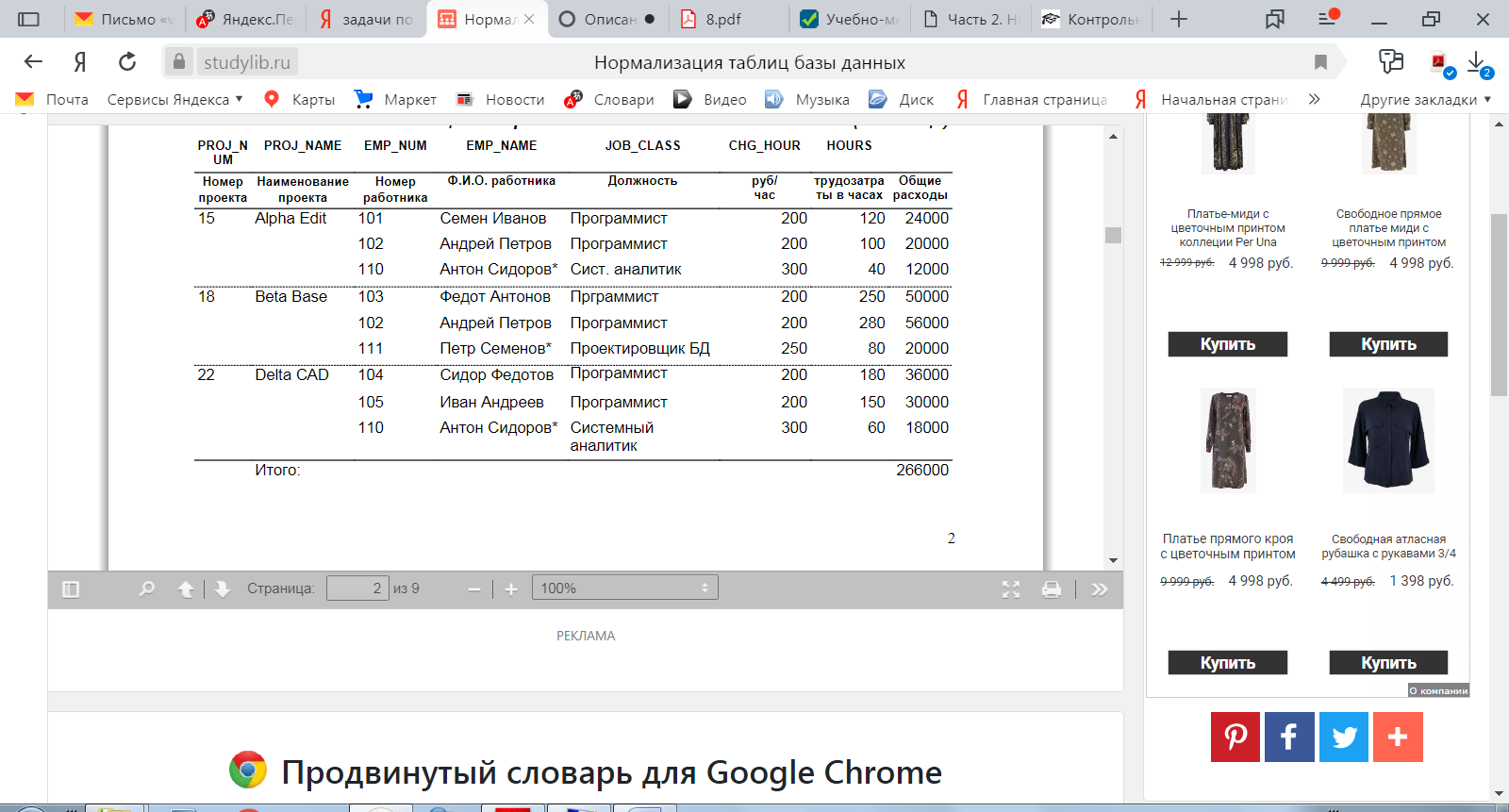
*Вариант 2 – отношение «Контрагенты»*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Наименование контрагента | Город | Адрес | Вид контрагента | Должность контактного лица | Ф.И.О.  контактного лица | Код города | Телефон |
| Поршневой завод | Владимир | ул. Кольцевая, 17 | Поставщик | зам. дир. | Иванов И.И. | 3254 | 76-15-95 |
| Поршневой завод | Владимир | ул. Кольцевая, 17 | Поставщик | нач. отд. сбыта | Петров П.П. | 3254 | 76-15-35 |
| ООО «Вымпел» | Курск | ул. Гоголя, 25 | Клиент, Поставщик | директор | Сидоров С.С. | 7634 | 66-65-38 |
| ИП «Альфа» | Владимир | ул.Пушкинская, 37 | Клиент, Поставщик | директор | Васильев В.В. | 3254 | 74-57-45 |

*Вариант 3 – отношение «Отдел кадров»*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код сотрудника | ФИО | Должность | Номер отдела | Наименование отдела | Квалификация |
| 7513 | Иванов И.И | Программист | 120 | Отдел проектирования | С, Java |
| 9842 | Петров А.А. | Администратор БД | 30 | Финансовый отдел | MS SQL Server |
| 6651 | Сорокин А.П. | Прогрсммист | 120 | Отдел проектирования | VB, Java |
| 9006 | Ворнов Г.Р. | Системный администраторо | 120 | Отдел проектирования | Windows, Linux |

*Вариант 4 – отношение «Ведомость расходов»*



*Вариант 3 – отношение «Отдел кадров»*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код сотрудника | ФИО | Должность | Номер отдела | Наименование отдела | Квалификация |
| 7513 | Иванов И.И | Программист | 120 | Отдел проектирования | С, Java |
| 9842 | Петров А.А. | Администратор БД | 30 | Финансовый отдел | MS SQL Server |
| 6651 | Сорокин А.П. | Прогрсммист | 120 | Отдел проектирования | VB, Java |
| 9006 | Ворнов Г.Р. | Системный администраторо | 120 | Отдел проектирования | Windows, Linux |

Сотрудники

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код сотрудника | ФИО | Должность | Номер отдела | Квалификация |
| 7513 | Иванов И.И | 1 | 120 | С, Java |
| 9842 | Петров А.А. | 2 | 30 | MS SQL Server |
| 6651 | Сорокин А.П. | 1 | 120 | VB, Java |
| 9006 | Ворнов Г.Р. | 3 | 120 | Windows, Linux |

Должности

|  |  |
| --- | --- |
| Код должности | Должность |
| 1 | Программист |
| 2 | Администратор БД |
| 3 | Системный администратор |

Отделы

|  |  |
| --- | --- |
| Номер отдела | Наименование отдела |
| 120 | Отдел проектирования |
| 30 | Финансовый отдел |
| 120 | Отдел проектирования |
| 120 | Отдел проектирования |

*Вариант 1 – отношение «Морские перевозки»*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер судна | Название | Номер рейса | Дата погрузки | Порт погрузки | Дата прибытия | Порт прибытия | Ф.И.О.  капитана | Вид судна | Грузо  подъем  ность, тонны |
| 526 | Japan Bear | 9201W | 5/31/92 | SFO | 6/6/92 | HNL | Емелин А.О. | Сухогруз | 500 |
| 603 | Korea Bear | 9202W | 5/05/92 | OAK | 6/19/92 | OSA | Крылов О.Б. | Ролкер | 1000 |
| 531 | China Bear | 9203W | 6/20/92 | LAX | 7/10/92 | PAP | Мухин Е.А. | Универсал | 1500 |
| 526 | Japan Bear | 9204W | 8/20/92 | SFO | 8/27/92 | HNL | Емелин А.О. | Сухогруз | 500 |

Судно

|  |  |  |  |
| --- | --- | --- | --- |
| Номер судна | Название | Вид судна | Грузо  подъем  ность, тонны |
| 526 | Japan Bear | 1 | 500 |
| 603 | Korea Bear | 2 | 1000 |
| 531 | China Bear | 3 | 1500 |

Виды судна

|  |  |
| --- | --- |
| Код вида | Вид |
| 1 | Сухогруз |
| 2 | Ролкер |
| 3 | Универсал |

Капитаны

|  |  |
| --- | --- |
| Код | Ф.И.О.  капитана |
| 1 | Емелин А.О. |
| 2 | Крылов О.Б. |
| 3 | Мухин Е.А. |

Рейсы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер рейса | Дата погрузки | Порт погрузки | Дата прибытия | Порт прибытия | Номер судна | Код  капитана |
| 9201W | 5/31/92 | SFO | 6/6/92 | HNL | 526 | 1 |
| 9202W | 5/05/92 | OAK | 6/19/92 | OSA | 603 | 2 |
| 9203W | 6/20/92 | LAX | 7/10/92 | PAP | 531 | 3 |
| 9204W | 8/20/92 | SFO | 8/27/92 | HNL | 526 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер судна | Название | Номер рейса | Дата погрузки | Порт погрузки | Дата прибытия | Порт прибытия | Ф.И.О.  капитана | Вид судна | Грузо  подъем  ность, тонны |
| 526 | Japan Bear | 9201W | 5/31/92 | SFO | 6/6/92 | HNL | Емелин А.О. | Сухогруз | 500 |
| 603 | Korea Bear | 9202W | 5/05/92 | OAK | 6/19/92 | OSA | Крылов О.Б. | Ролкер | 1000 |
| 531 | China Bear | 9203W | 6/20/92 | LAX | 7/10/92 | PAP | Мухин Е.А. | Универсал | 1500 |
| 526 | Japan Bear | 9204W | 8/20/92 | SFO | 8/27/92 | HNL | Емелин А.О. | Сухогруз | 500 |

|  |  |  |  |
| --- | --- | --- | --- |
| Номер судна | Название судна | Код вида | Код грузоподъемности |
| 526 | Japan Bear | 1 | 1 |
| 603 | Korea Bear | 2 | 2 |
| 531 | China Bear | 3 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код вида | Вид |  | Код грузоподъемности | Грузоподъемность |
| 1 | Сухогруз | 1 | 500 |
| 2 | Ролкер | 2 | 1000 |
| 3 | Универсал | 3 | 1500 |

|  |  |
| --- | --- |
| Код капитана | Ф.И.О.  капитана |
| 1 | Емелин А.О. |
| 2 | Крылов О.Б. |
| 3 | Мухин Е.А. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер рейса | Дата погрузки | Порт погрузки | Дата прибытия | Порт прибытия | Код капитана | Номер судна |
| 9201W | 5/31/92 | SFO | 6/6/92 | HNL | 1 | 526 |
| 9202W | 5/05/92 | OAK | 6/19/92 | OSA | 2 | 603 |
| 9203W | 6/20/92 | LAX | 7/10/92 | PAP | 3 | 531 |
| 9204W | 8/20/92 | SFO | 8/27/92 | HNL | 1 | 526 |