ИНСТИТУТ ИНФОРМАЦИОННЫХ И ВЫЧИСЛИТЕЛЬНЫХ ТЕХНОЛОГИЙ

Кафедра Вычислительных Технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ТИПОВОМУ РАСЧЕТУ**

**по дисциплине «Базы данных»**

Выполнил: студент Швердяков М.А.

Группа: А-06-21

Проверил: профессор Бородин Г.А.

**Москва, 2023**

Оглавление

[1. Цель типового расчета 3](#_Toc90125433)

[2. Область применения моделируемой базы данных 3](#_Toc90125434)

[3. Нормализация исходной базы данных 3](#_Toc90125435)

[3.1. Ненормализованная исходная база данных 3](#_Toc90125436)

[3.2. Нормализация до первой нормальной формы 4](#_Toc90125437)

[3.3. Нормализация до второй нормальной формы 5](#_Toc90125438)

[3.4. Нормализация до третьей нормальной формы 6](#_Toc90125439)

[4. Типы связей в моделируемой базе данных 7](#_Toc90125440)

[4.1. Обоснование выбора типа связей 7](#_Toc90125441)

[4.2. Опции ссылочной целостности базы данных 8](#_Toc90125441)

[5. Домен и ограничения 9](#_Toc90125443)

[6. Физическая модель 10](#_Toc90125444)

[7. Представление 10](#_Toc90125445)

[8. Результаты моделирования базы данных 11](#_Toc90125446)

[9. Оценка размеров базы данных 13](#_Toc90125447)

# 1. Цель типового расчета

Целью типового расчета является практическое освоение основ моделирования реляционных баз данных с помощью CASE-средств.

# 2. Область применения моделируемой базы данных

Моделируемая база данных применима для хранения и систематизирования информации о клиентах компании по продаже оптовых товаров. В базе находится информация о клиентах компании, о информации для предоставления счета (название компании, менеджере по закупкам, ИНН, адрес), контактной информации (адрес электронной почты), информации для повышения KPI менеджера по продажам (дата последнего звонка) и дополнительной информации (отсрочка платежа и скидка клиента от РРЦ). С помощью данной модели базы данных можно упростить работу менеджеру и руководителю отдела продаж любой оптовой компании по отслеживанию своих клиентов.

# 3. Нормализация исходной базы данных

## 3.1. Ненормализованная исходная база данных

Ненормализованная исходная база данных представлена на рисунке 1.

   
Рисунок 1 — Ненормализованная база данных

Данная база данных является однотабличной и обладает избыточностью, что влечет за собой несогласованность данных, повторение имеющейся информации и потерю времени при изменении базы данных. Так, при изменении любой информации о клиенте придется менять все записи, содержащие эту информацию. Также, несогласованность записей приводит к тому, что одинаковые данные, могут быть восприняты как разные. К примеру название одной и той же компании записанное в разном регистре, будет воспринято как разные компании.

Для решения данной проблемы необходимо нормализировать исходную базу данных

## 3.2. Нормализация до первой нормальной формы

Требования первой нормальной формы:

1. Отсутствие повторяющихся данных
2. Все атрибуты содержат атомарные значения
3. Отсутствие повторяющихся групп атрибутов
4. Кортежи и атрибуты не упорядочены

В базе данных, представленной на рисунке 1, атрибут «Manager» не является атомарным. В нем хранится информация о менеджере по продажам, ответственном за заказ. Наличие неатомарных атрибутов не соответствует первой нормальной форме. Для решения данной проблемы, разобьем этот атрибут на два новых атрибута: «Manger\_Name» и «Manager\_Surname». Для того, чтобы исключить возможность повторяющихся данных необходимо к этим двум атрибутам добавить еще один «Manger\_ID».

Заметим, что один клиент может иметь много заказов, поэтому для начала нормализации необходимо разбить исходную базу данных на две сущности: «Clients» и «Orders», первая из которых будет содержать информацию о оптовых клиентах компании, а вторая — о деталях заказа.

Для сущности «Clients» определим первичный ключ «Client\_ID», а для сущности «Orders» составной первичный ключ из атрибутов «Order\_ID» и «Client\_ID».

База данных, нормализованная к первой нормальной форме, представлена на рисунке 2.



Рисунок 2 — База данных, нормализованная до первой нормальной формы

## 3.3. Нормализация до второй нормальной формы

Требования второй нормальной формы:

1. База данных удовлетворяет требованиям первой нормальной формы
2. Любой неключевой атрибут однозначно идентифицируется полным набором ключевых атрибутов

Обратим внимание на сущность «Orders». Атрибуты «Manager\_ID», «Manager\_Name» и «Manager\_Surname», не зависят от ключевого атрибута «Order\_ID». По этой причине их необходимо перенести в новую сущность под названием «Manager», с первичным ключом «Manager\_ID», чтобы удовлетворить требованиям второй нормальной формы.

В результате получим базу данных, нормализованную до второй нормальной формы, которая представлена на рисунке 3.



Рисунок 3 — База данных, нормализованная до второй нормальной формы

## 3.4. Нормализация до третьей нормальной формы

Требования третьей нормальной формы:

1) База данных удовлетворяет требованиям второй нормальной формы

2) Ни один из неключевых атрибутов не идентифицируется с помощью другого неключевого атрибута

Заметим, что в сущности «Clients» атрибут «Discount\_Percent» зависит от неключевого атрибута «Discount\_ID». Чтобы удовлетворить всем требованиям третьей нормальной формы, необходимо все вышеперечисленные атрибуты перенести в новую сущность «Discount», где атрибут «Discount\_ID» назначить первичным ключом.

Результатом данных действий является база данных, нормализованная до третьей нормальной формы и представленная на рисунке 4.



Рисунок 4 — База данных, нормализованная до третьей нормальной формы

# 4. Типы связей в моделируемой базе данных

## 4.1. Обоснование выбора типа связей

1) Связь Clients-to-Orders.

Выбрана идентифицирующая связь один-ко-многим, так как клиент определяет заказ, и у одного клиента может быть несколько заказов

2) Связь Discount-to-Clients.

Выбрана неидентифицирующая связь один-ко-многим с необязательным участием, так как скидка не определяет клиента, то есть сущности являются независимыми. Выбираем связь один-ко-многим, так как одна скидка может распространяться на несколько клиентов, но у одного клиента не может быть несколько скидок. Пример: компания дает приветственную скидку всем новым клиентам.

3) Связь Manager-to-Orders.

Выбрана неидентифицирующая связь один-ко-многим с обязательным участием, поскольку заказ не идентифицируется с помощью менеджера по продажам. Выбрана связь один-ко-многим, так как у одного менеджера может быть несколько заказов, но у одного заказа не может быть несколько менеджеров. Пример: менеджер по продажам Иван ведет три заказа, но у него не хватает времени, поэтому он передает один из заказов другому менеджеру. Обязательное участие, так как заказ должен быть привязан к менеджеру.

### 4.2. Опции ссылочной целостности базы данных

Имеется три связи: «Client\_to\_Orders», «Discount\_to\_Client», «Manager\_to\_Orders». Рассмотрим каждую связь подробнее.

Во всех связях для «Parent Update» выбрано «Restrict», так как сущности связаны только первичными ключами.

Для связи «Client\_to\_Orders» для «Parent Delete» выбрано «Cascade», так как после удаления пользователя, нет никакой необходимости хранить информацию о его заказах. Для «Child Insert» выбрано «Restrict», так как заказ без клиента невозможен.

Для связи «Discount\_to\_Client» для «Child Insert» выбираем «Set Null», так как у клиента может не быть скидки. Для «Parent Delete» выбрано «Set Null», так, например, если скидка была всего на месяц, то после окончания срока скидка исчезает.

Для связи «Manager\_to\_Orders» для «Child Insert» выбираем «Restrict», так как нельзя создать заказ без привязки к менеджеру. Для «Parent Delete» выбрано «Set Null», например, если менеджер увольняется, какое-то время заказ будет существовать без менеджера.

В результате получим базу данных с настроенной ссылочной целостностью. Она представлена на рисунке 5.



Рисунок 5 — База данных с настроенной ссылочной целостностью

# 

# 5. Домен и ограничения

Рассмотрим атрибут «Discount\_Percent», который обозначает процент скидки клиента. Используем домен «Percent», чтобы ограничить это значение в пределах ста.

Ограничение «Percent» представлено в таблице 1. Применение пользовательского домена в базе данных представлено на рисунке 6.

Таблица 1 — Ограничение на значение скидки

| **Used by Column(s) of "Percent" Domain** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Name** | | | **Datatype** | | |
| Percent | | | float | | |
| **Validation(s) of "Percent" Used by Column** | | | | | |
| **Name** | **Type** | **Min** | | **Max** | **Rule** |
| Min\_Max | Client/Server | 1 | | 100 | @col BETWEEN 1 AND 100 |



Рисунок 6 — Применение пользовательского домена «Percent»

# 

# 6. Физическая модель

Для создания базы данных была выбрана СУБД SQL Server. Физическая модель базы данных представлена на рисунке 7.



Рисунок 7 — Физическая модель базы данных

# 7. Представление

Представление, содержащее основную информацию, необходимую для логистической компании, доставляющей заказ, а именно название компании клиента, ИНН, дату доставки и количество товаров в посылке. продемонстрированное на рисунке 8.



Рисунок 8 — Представление

# 8. Результаты моделирования базы данных

Сведения о сущностях, атрибутах, ключах и триггерах представлены в таблице 2.

Таблица 2 — Сведения о базе данных

|  |  |  |  |
| --- | --- | --- | --- |
| Table | | | |
| Name | | | |
| Discount | | | |
| Manager | | | |
| Clients | | | |
| Orders | | | |
| «Discount» Table | | | |
| Name | Is PK | Is FK | Domain |
| Discount\_ID | Yes | No | Number |
| Discount\_Percent | No | No | Percent |
| Child Table Name | Referential Integrity | | |
| Clients | Parent Delete | Parent Update | Child Insert |
| Set Null | Restrict | Set Null |
| «Manager» Table | | | |
| Name | Is PK | Is FK | Domain |
| Manager\_ID | Yes | No | Number |
| Manager\_Name | No | No | String |
| Manager\_Surname | No | No | String |
| Child Table Name | Referential Integrity | | |
| Orders | Parent Delete | Parent Update | Child Insert |
| Set Null | Restrict | Restrict |
| «Orders» Table | | | |
| Name | Is PK | Is FK | Domain |
| Order\_ID | Yes | No | Number |
| Client\_ID | Yes | Yes | Number |
| Shipment\_Date | No | No | String |
| Wholesale\_Price | No | No | Number |
| MSRP | No | No | Number |
| Quantity\_of\_Goods | No | No | Number |
| Order\_Date | No | No | Datetime |
| Manager\_ID | No | Yes | Number |
| Parent Table Name | Referential Integrity | | |
| Clients | Parent Delete | Parent Update | Child Insert |
| Cascade | Restrict | Restrict |
| Parent Table Name | Referential Integrity | | |
| Manager | Parent Delete | Parent Update | Child Insert |
| Set Null | Restrict | Restrict |
| «Clients» Table | | | |
| Name | Is PK | Is FK | Domain |
| Client\_ID | Yes | No | Number |
| Client\_Company\_Name | No | No | String |
| FICA | No | No | Number |
| City | No | No | String |
| Last\_Call | No | No | Datetime |
| Email | No | No | String |
| Discount\_ID | No | Yes | Number |
| Payment\_Delay | No | No | Datetime |
| Child Table Name | Referential Integrity | | |
| Orders | Parent Delete | Parent Update | Child Insert |
| Cascade | Restrict | Restrict |
| Parent Table Name | Referential Integrity | | |
| Discount | Parent Delete | Parent Update | Child Insert |
| Set Null | Restrict | Set Null |

# 

# 9. Оценка размеров базы данных

Проведем оценку размеров смоделированной базы данных. Информация о начальном количестве записей каждой сущности и его ежемесячном приросте:

Сущность «Clients» — 100. Ежемесячный прирост — 5.

Сущность «Orders» — 200. Ежемесячный прирост — 15.

Сущность «Discount» — 10. Ежемесячный прирост — 1.

Сущность «Manager» — 5. Ежемесячный прирост — 1.

Оценки размеров базы данных на разный период времени представлены на рисунке 9, рисунке 10 и рисунке 11.

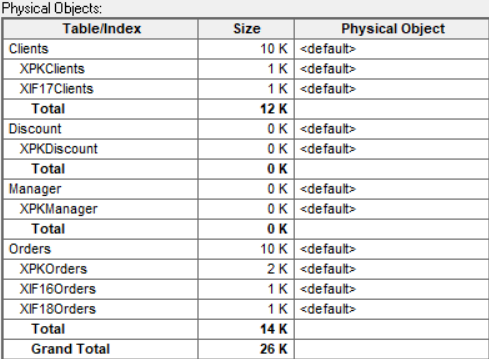


Рисунок 9 — Оценка размеров базы данных через три месяца

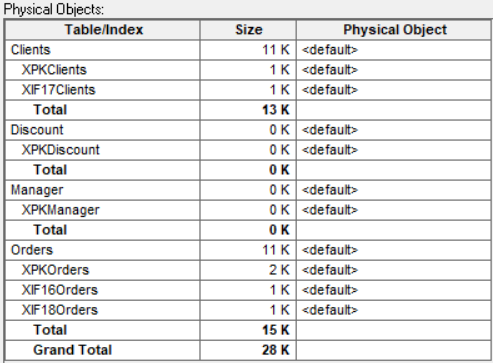


Рисунок 10 — Оценка размеров базы данных через полгода

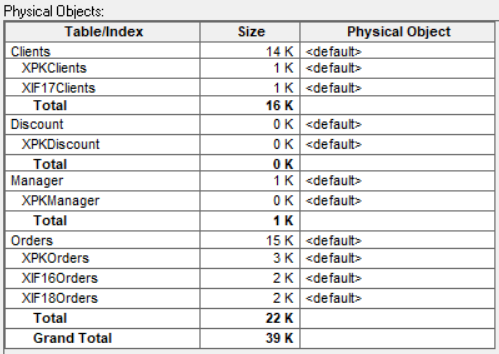


Рисунок 11 — Оценка размеров базы данных через год