Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 1

по курсу «Программно-аппаратные средства хранения и ОД»

на тему «Проектирование базы данных с использованием ER-технологии»

Вариант 2

Выполнили

студенты группы 22ВВП1:

Зубриянова А.А.

Кондратьева В.И.

Приняли:

Дубинин В.Н.

Карамышева Н.С.

Пенза 2025

**Цель работы**

Целью данного проекта является разработка базы данных для управления гостиницей с использованием ER-технологии. В системе должны быть реализованы основные процессы бронирования номеров, учёта клиентов, предоставления скидок.

**Лабораторное задание**

При выполнении лабораторной работы необходимо:

* для своего варианта, соответствующего определенной предметной области, построить логическую модель данных в соответствии со стандартом IDEF1X;
* построить физическую модель;
* составить отчет по лабораторной работе.

*Вариант 2: Гостиница*

**Теоретическая часть**

Этапы построения БД:

1. Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) и [модель данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).
2. Логическое (даталогическое) проектирование — создание [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) на основе конкретной [модели данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), например, [реляционной модели данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Для реляционной модели данных даталогическая модель — набор схем [отношений](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)), обычно с указанием [первичных ключей](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B2%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BB%D1%8E%D1%87), а также «связей» между отношениями, представляющих собой [внешние ключи](https://ru.wikipedia.org/wiki/%D0%92%D0%BD%D0%B5%D1%88%D0%BD%D0%B8%D0%B9_%D0%BA%D0%BB%D1%8E%D1%87).
3. Физическое проектирование — создание [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) для конкретной [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94). Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

Нормальная форма — свойство [отношения](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)) в [реляционной модели данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Нормализация предполагает применение нормальных форм к структуре данных. Существуют 7 нормальных форм. Каждая нормальная форма (за исключением первой) подразумевает, что к данным уже была применена предыдущая нормальная форма. Например, прежде чем применить третью нормальную форму к данным должна быть применена вторая нормальная форма. И строго говоря, база данных считается нормализованной, если к ней применяется третья нормальная форма и выше.

Первая нормальная форма (1NF) предполагает, что сохраняемые данные на пересечении строк и столбцов должны представлять скалярное значение, а таблицы не должны содержать повторяющихся строк.

Вторая нормальная форма (2NF) предполагает, что каждый столбец, не являющийся ключом, должен зависеть от первичного ключа.

Третья нормальная форма (3NF) предполагает, что каждый столбец, не являющийся ключом, должен зависеть только от первичного ключа.

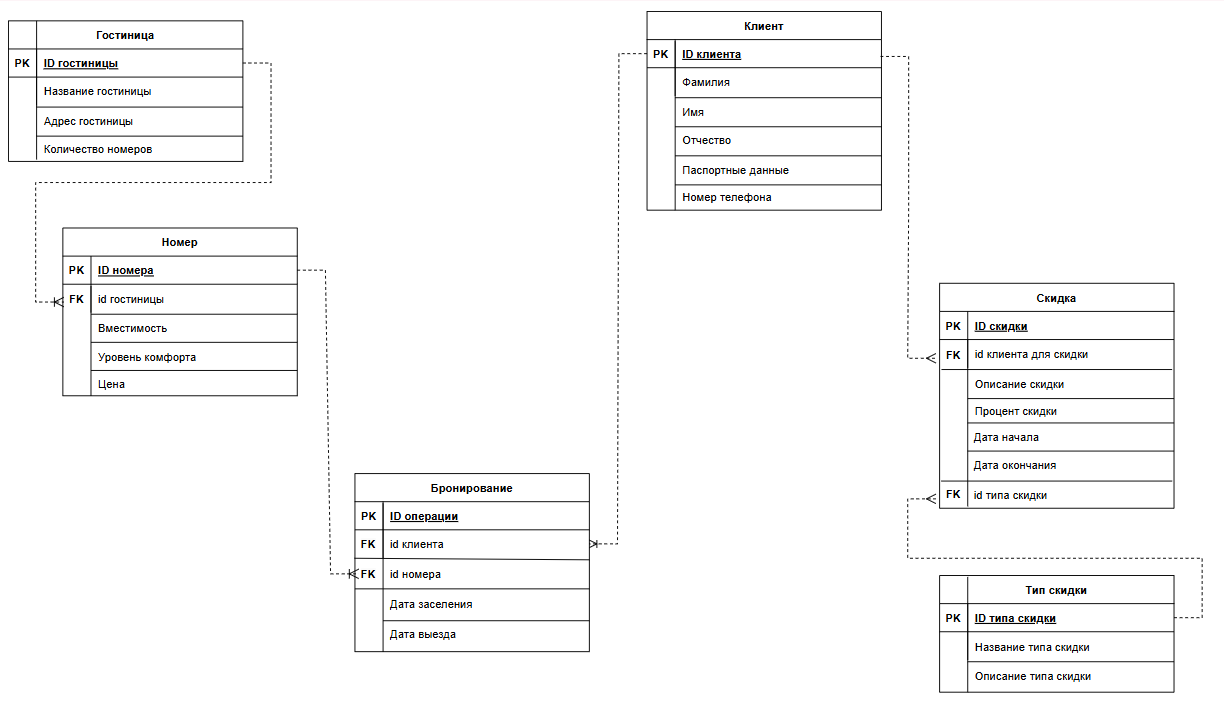
Нормальная форма Бойса-Кодда (BCNF) является немного более строгой версией третьей нормальной формы.

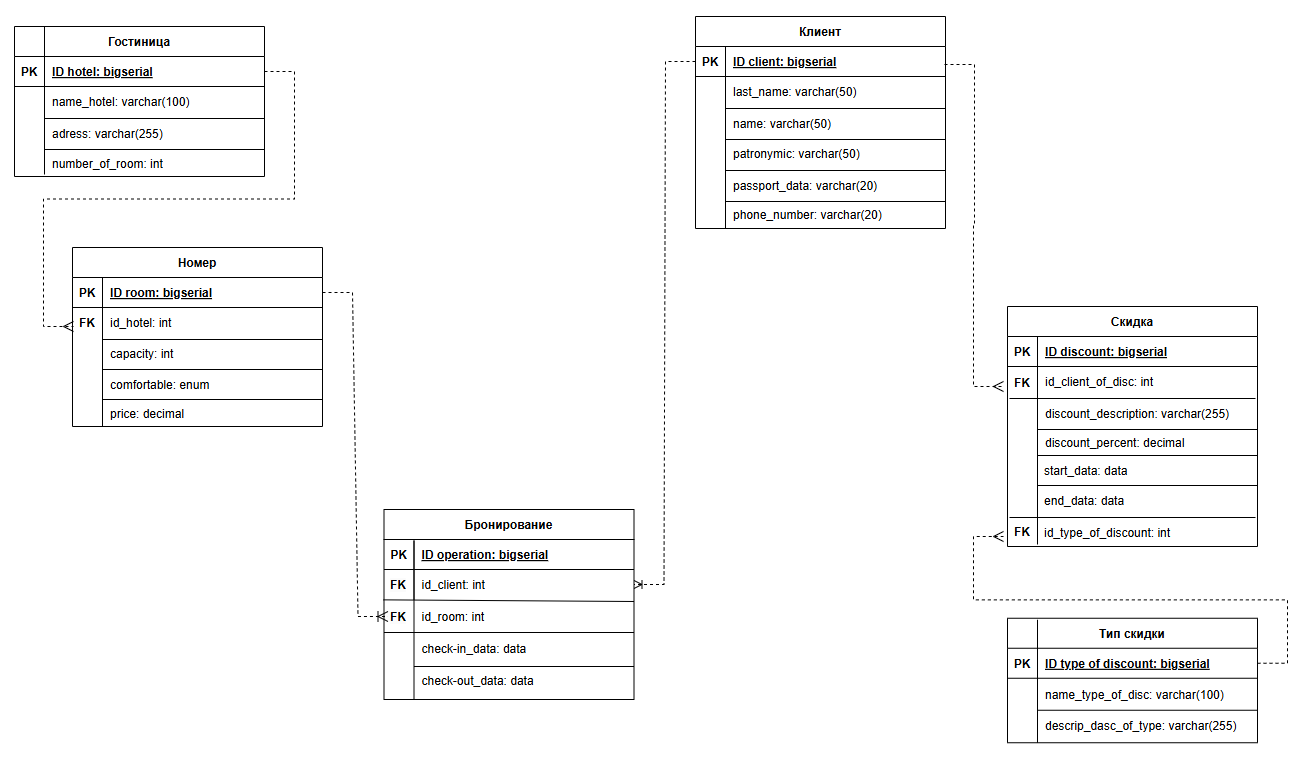
Четвертая нормальная форма (4NF) применяется для устранения многозначных зависимостей (multivalued dependencies) - таких зависимостей, где столбец с первичным ключом имеет связь один-ко-многим со столбцом, который не является ключом. Эта нормальная форма устраняет некорректные отношения многие-ко-многим.

Пятая нормальная форма (5NF) разделяет таблицы на более малые таблицы для устранения избыточности данных. Разбиение идет до тех пор, пока нельзя будет воссоздать оригинальную таблицу путем объединения малых таблиц.

Шестая нормальная форма (domain key normal form / 6NF). Каждое ограничение в связях между таблицами должно зависеть только от ограничений ключа и ограничений домена, где домен представляет набор допустимых значений для столбца. Эта форма предотвращает добавление недопустимых данных путем установки ограничения на уровне отношений между таблицами, но не на уровне таблиц или столбцов. Данная форма, как правило, не применима на уровне СУБД, в том числе и в SQL Server.

**1. Логическая и физическая модель предметной области**





**2. Таблица сущностей**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название сущности | Количество | Изменение количества сущностей | Идентификатор | Ограничение доступа | Тип связи |
| Гостиница | 1 | 0% | Hotel | Администратор, владелец | 1:M (Номер) |
| Номер | 90 | 10% | Room | Администратор, владелец | M:1 (Гостиница), 1:M (Бронирование) |
| Клиент | 1000 | 15% | Client | Администратор | 1:M (Бронирование) |
| Бронирование | 10000 | 20% | Booking | Администратор | M:1 (Клиент), M:1 (Номер) |
| Скидка | 100 | 5% | Discount | Администратор, владелец | 1:M (Клиент) |
| Тип скидки | 5 | 2% | DiscountType | Администратор, владелец | 1:M (Скидка) |

**3. Таблицы атрибутов**

Таблица атрибутов сущности "Гостиница"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID гостиницы | BIGSERIAL | 1 | Администратор, владелец | Ключ |
| Название гостиницы | VARCHAR(100) | - | Владелец |  |
| Адрес гостиницы | VARCHAR(255) | - | Владелец |  |
| Количество номеров | INT | 1-100 | Администратор, владелец |  |

Таблица атрибутов сущности "Номер"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID номера | BIGSERIAL | 1-90 | Администратор | Ключ |
| id гостиницы | INT | 1 | Администратор, владелец | Внешний Ключ |
| Вместимость | INT | 1-5 | Администратор |  |
| Уровень комфорта | ENUM | Люкс, Полулюкс, Обычный | Администратор |  |
| Цена | DECIMAL | 500-5000 | Администратор, владелец |  |

Таблица атрибутов сущности "Клиент"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID клиента | BIGSERIAL | 1-1000 | Администратор | Ключ |
| Фамилия | VARCHAR(50) | - | Администратор |  |
| Имя | VARCHAR(50) | - | Администратор |  |
| Отчество | VARCHAR(50) | - | Администратор |  |
| Паспортные данные | VARCHAR(20) | - | Администратор |  |
| Номер телефона | VARCHAR(20) | - | Администратор |  |

Таблица атрибутов сущности "Бронирование"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID операции | BIGSERIAL | 1-10000 | Администратор | Ключ |
| id клиента | INT | 1-1000 | Администратор | Внешний ключ |
| id номера | INT | 1-90 | Администратор | Внешний ключ |
| Дата заселения | DATE | - | Администратор |  |
| Дата выезда | DATE | - | Администратор |  |

Таблица атрибутов сущности "Скидка"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID скидки | BIGSERIAL | 1-50 | Администратор | Ключ |
| id клиента для скидки | INT | 1-1000 | Администратор | Внешний ключ |
| Описание скидки | VARCHAR(255) | - | Администратор, владелец |  |
| Процент скидки | DECIMAL | 1-100 | Администратор, владелец |  |
| Дата начала | DATE | - | Администратор |  |
| Дата окончания | DATE | - | Администратор |  |
| id типа скидки | INT | 1-5 | Администратор | Внешний ключ |

Таблица атрибутов сущности "Тип скидки"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID типа скидки | BIGSERIAL | 1-5 | Администратор | Ключ |
| Название скидки | VARCHAR(100) | Постоянный клиент, Ветеран, Корпоративная, Сезонная, Промо | Администратор, владелец |  |
| Описание типа скидки | VARCHAR(255) | - | Администратор |  |

**4. Список запросов**

1. Найти список номеров с заданными параметрами (категория, вместимость, цена).
2. Получить список всех бронирований определенного клиента.
3. Найти все номера, которые были забронированы в течение последнего месяца.
4. Выдать список клиентов, имеющих скидки.
5. Найти скидки, действовавшие в определённый период времени.
6. Выдать список номеров категории "Люкс", доступных на заданную дату.

**Вывод**

В ходе проектирования базы данных была создана ER-модель, описывающая основные сущности и их связи. Данная модель позволяет эффективно управлять бронированием номеров, учётом клиентов и системой скидок. Реализация базы данных в реляционной форме обеспечит удобство работы с информацией и возможность анализа данных для оптимизации бизнес-процессов гостиницы. Разработанная система может быть расширена и адаптирована под специфические потребности конкретной гостиницы.