

---

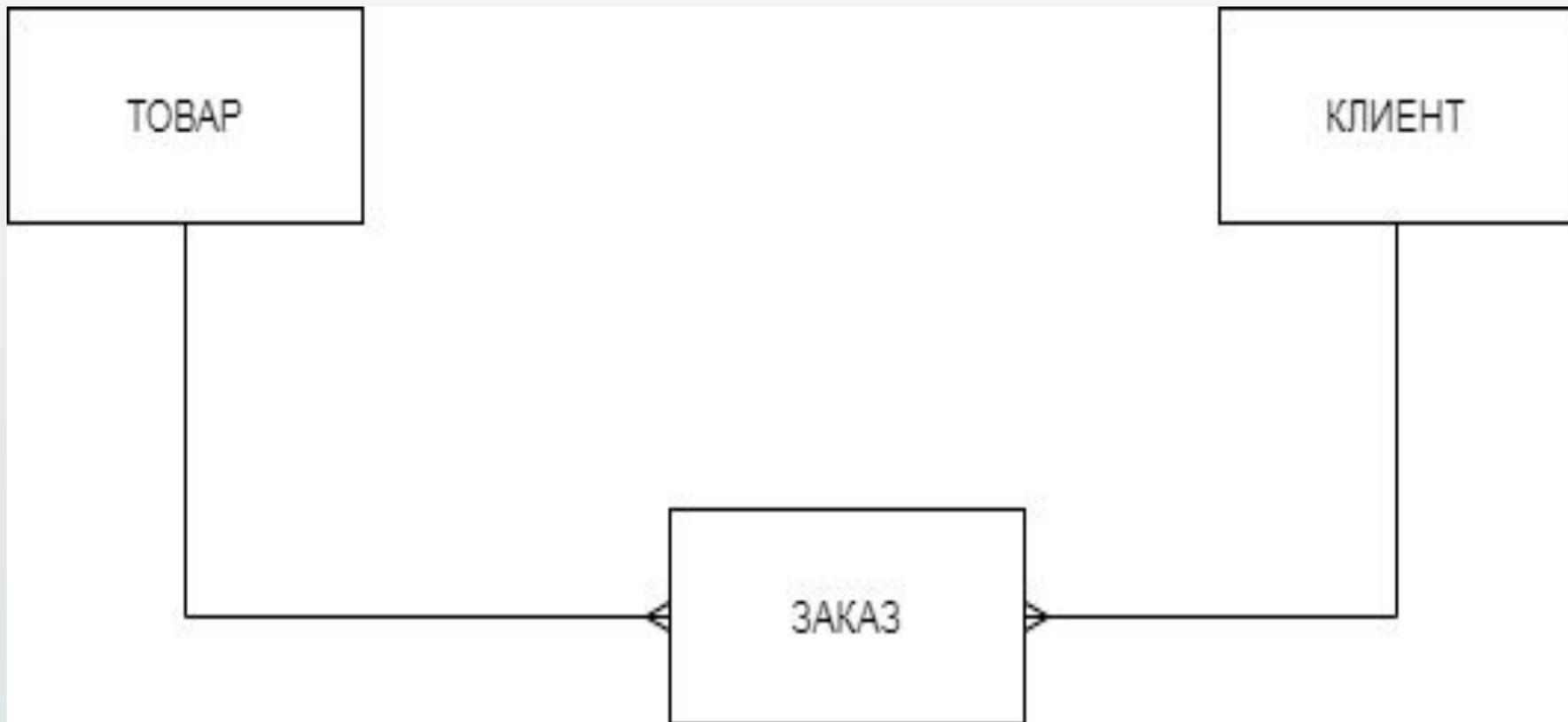
# Работа с MySQL



# Проектирование базы данных Sales

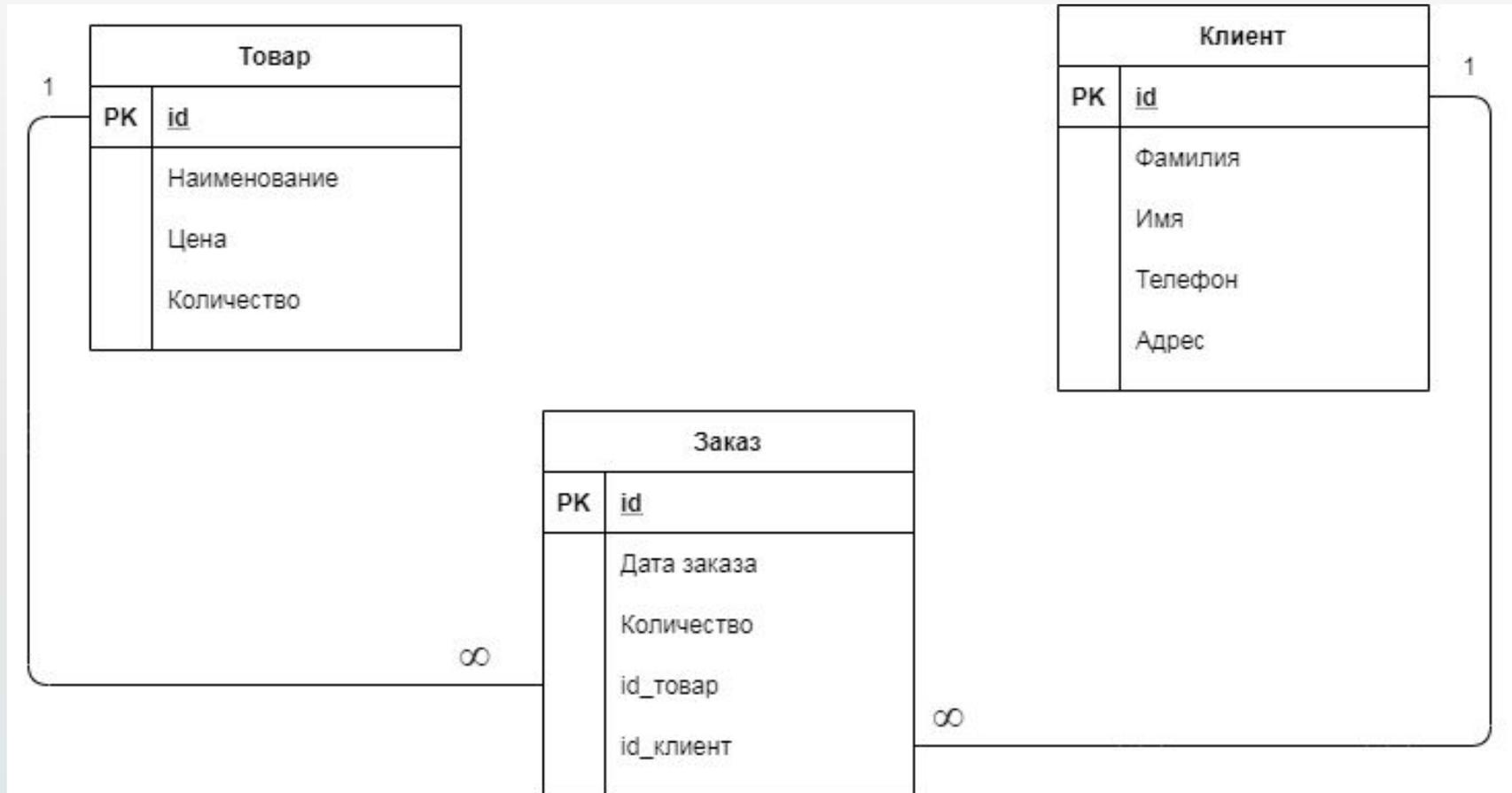
---

## Концептуальная модель



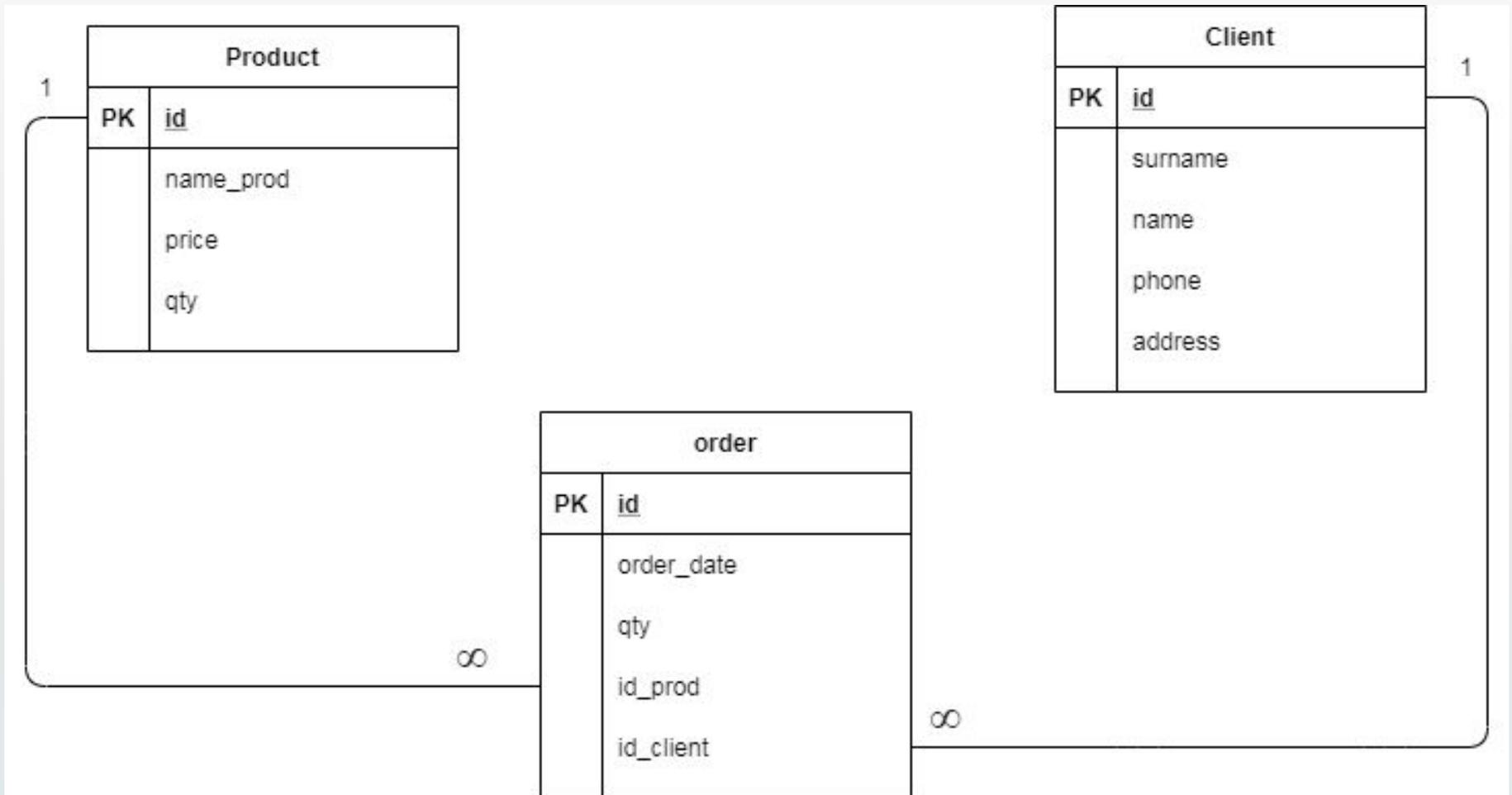
# Проектирование базы данных Sales

## Логическая модель



# Проектирование базы данных Sales

## Логическая модель



# Создание базы данных

---

*Синтаксис:*

```
CREATE DATABASE имя_БД  
[CHARACTER SET имя_кодировки]  
[COLLATE имя_правила_сравнения]
```

Кодировка, указанная при создании базы данных, будет по умолчанию использоваться для таблиц этой базы данных.

Если в базу данных необходимо импортировать данные, которые находятся в кодировке utf8, то при создании базы данных нужно указать:

```
CREATE DATABASE sales  
CHARACTER SET utf8  
COLLATE utf8_general_ci;
```

---



# Создание таблицы базы данных

---

Синтаксис:

```
CREATE TABLE имя_таблицы  
(<Имя_столбца1>  
  <Тип_столбца1>[<Свойства_столбца1>],  
<Имя_столбца2>  
  <Тип_столбца2>[<Свойства_столбца2>],  
...  
[<Информация о ключевых полях и индексах>])  
[<Опциональные свойства таблицы>];
```

---



# Первичный ключ

---

*Синтаксис:*

**PRIMARY KEY (<имя ключа>)**



# Создание таблицы Product

---

## Пример:

```
CREATE TABLE Product  
(id SERIAL,  
name_prod VARCHAR(40),  
price DECIMAL(8,2),  
qty INT UNSIGNED,  
PRIMARY KEY(id))  
  
ENGINE InnoDB CHARSET utf8;
```

---



# Создание таблицы Client

---

## Пример:

```
CREATE TABLE Client  
(id SERIAL,  
Surname VARCHAR(40),  
Name VARCHAR(20),  
Phone VARCHAR(20),  
Address VARCHAR(100),  
PRIMARY KEY(id))  
  
ENGINE InnoDB CHARSET utf8;
```

---



# Внешний ключ

Синтаксис:

**FOREIGN KEY** (<имя ключа>)**REFERENCES** <имя главной таблицы>(<имя первичного ключа главной таблицы>)

[правила целостности данных]

Правила поддержки целостности данных:

1. **ON UPDATE |DELETE CASCADE** –каскадное обновление или удаление строк дочерней таблицы
2. **ON UPDATE |DELETE SET NULL**- обнуление значения внешнего ключа в соответствующих строках дочерней таблицы.
3. **ON UPDATE |DELETE RESRRICT | NO ACTION** – запрет обновления или удаления строк родительской таблицы при наличии ссылающихся строк дочерней таблицы

# Создание таблицы Order

---

## Пример:

```
CREATE TABLE Order
(id SERIAL,
Order_date DATE,
qty INT UNSIGNED,
id_Product BIGINT UNSIGNED NOT NULL,
id_Client BIGINT UNSIGNED NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY(id_product) REFERENCES Product (id) ON
UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY(id_client) REFERENCES Client(id) ON UPDATE
CASCADE ON DELETE RESTRICT)
ENGINE InnoDB CHARSET utf8;
```

---



# Изменение структуры таблицы

Синтаксис:

## Переименовать таблицу

**ALTER TABLE** <Имя таблицы> **RENAME** <новое имя таблицы>;

## Изменить тип столбца

**ALTER TABLE** <Имя таблицы> **MODIFY** <Имя столбца> <новый тип данных> ; ( без изменения имени столбца)

**ALTER TABLE** <Имя таблицы> **CHANGE** <прежнее имя столбца> <новое имя столбца> <новый тип данных> [Свойства] ;  
( с изменением имени столбца)

## Удалить столбец

**ALTER TABLE** <Имя таблицы> **DROP** <Имя столбца> ;

# Изменение структуры таблицы

---

Синтаксис:

## Добавление столбца

```
ALTER TABLE <Имя таблицы> ADD <Имя столбца> <Тип столбца> [Свойства] [FIRST / AFTER <Имя предшествующего столбца>];
```

Пример: Добавить в таблицу **Client** поле **Возраст (age)**

```
ALTER TABLE Client
```

```
ADD age INT
```

```
AFTER name;
```

---

# Изменение структуры таблицы

---

Синтаксис:

## Добавление первичного ключа

```
ALTER TABLE <Имя таблицы>  
ADD [CONSTRAINT <Имя ключа>  
PRIMARY KEY (Имя ключа);
```

где, *CONSTRAINT* –ограничения первичного ключа

## Добавление внешнего ключа

```
ALTER TABLE <Имя таблицы>  
ADD [CONSTRAINT <Имя ключа>  
FOREIGN KEY (Имя ключа) REFERENCES <имя главной  
таблицы>( <имя первичного ключа главной таблицы> )  
[правила целостности данных]
```