

№	Модель данных	Преимущества	Недостатки
1	Иерархическая модель	<ul style="list-style-type: none"> <li>- Проста и интуитивно понятна.</li> <li>- Эффективна для представления иерархической структуры данных.</li> <li>- Высокий уровень безопасности и целостности данных.</li> <li>- Минимальный расход памяти.</li> </ul>	<ul style="list-style-type: none"> <li>- Ограничения в представлении связей между данными.</li> <li>- Сложность при работе с данными, не соответствующими иерархической модели.</li> <li>- Не поддерживает отношения многие к многим.</li> </ul>
2	Сетевая модель	<ul style="list-style-type: none"> <li>- Позволяет эффективно описывать сложные отношения между данными.</li> <li>- Поддерживает более гибкую структуру данных, чем иерархическая модель.</li> <li>- Высокая эффективность затрат памяти.</li> <li>- Оперативность обработки данных.</li> </ul>	<ul style="list-style-type: none"> <li>- Сложность разработки и поддержки сетевых баз данных.</li> <li>- Зависимость от физической структуры данных.</li> <li>- Обилие понятий вариантов их взаимосвязей и особенностей реализации.</li> <li>- Ослабленный контроль целостности.</li> </ul>
3	Реляционная модель	<ul style="list-style-type: none"> <li>- Простота использования и понимания.</li> <li>- Стандартизированная модель с использованием SQL.</li> <li>- Обеспечивает целостность очень больших объемов данных</li> </ul>	<ul style="list-style-type: none"> <li>- Сложность при работе с комплексными структурами данных.</li> <li>- Проблемы производительности при больших объемах данных</li> <li>- Использует длительные и кратковременные блокировки, что</li> </ul>

			приводит к задержкам и недоступности данных.
4	Модель “сущность-связь”	<ul style="list-style-type: none"> <li>- Понятность и наглядность в описании структуры данных.</li> <li>- Помогает выделить ключевые сущности и связи между ними.</li> </ul>	<ul style="list-style-type: none"> <li>- Ограничения при описании сложных отношений между данными.</li> <li>- Не всегда покрывает все аспекты структуры данных.</li> </ul>
5	Семантическая модель	<ul style="list-style-type: none"> <li>- Уделяет внимание смыслу данных и их взаимосвязям.</li> <li>- Позволяет более точно определить семантическое значение данных.</li> </ul>	<ul style="list-style-type: none"> <li>- Сложность в разработке и поддержке такой модели.</li> <li>- Требуется дополнительных инструментов для полноценного использования.</li> </ul>
6	Объектно-ориентированная модель	<ul style="list-style-type: none"> <li>- Удобство работы с объектами и их методами.</li> <li>- Позволяет инкапсулировать данные и методы в объекты.</li> <li>- Позволяет создавать новые типы данных из уже существующих.</li> </ul>	<ul style="list-style-type: none"> <li>- Не всегда эффективно справляется с большим объемом данных.</li> <li>- Требуется обучения и понимания концепций ООП.</li> <li>- Отсутствие универсальной модели данных и общепризнанного языка запросов</li> </ul>
7	Объектно-реляционная модель	<ul style="list-style-type: none"> <li>- Сочетает преимущества реляционной и объектно-ориентированной моделей.</li> <li>- Позволяет работать как с табличными данными, так и с объектами.</li> </ul>	<ul style="list-style-type: none"> <li>- Сложность при интеграции и поддержке двух различных подходов.</li> <li>- Не всегда эффективна из-за дополнительных слоев абстракции.</li> </ul>

8	Полуструктурированная модель	<ul style="list-style-type: none"> <li>- Гибкость в представлении данных без фиксированной схемы.</li> <li>- Поддержка различных форматов данных, таких как JSON, XML и других.</li> </ul>	<ul style="list-style-type: none"> <li>- Ограничения в возможностях запросов данных.</li> <li>- Сложность в обеспечении целостности данных.</li> </ul>
---	------------------------------	--	--