

Реляционные СУБД (PostgreSQL, Oracle, Greenplum и т.д.)

Преимущества:

- Поддержка SQL
- ACID-транзакции гарантируют целостность данных
- Хорошо подходят для ETL-процессов и нормализованных моделей
- Поддерживают сложные JOIN'ы, агрегации, оконные функции
- Отлично работают с измерениями и фактами в моделях "звезда" и "снежинка"
- Greenplum – массово-параллельная СУБД (MPP), оптимизирована для аналитики

Недостатки:

- Могут быть медленны при обработке очень больших объёмов данных
 - Ограниченнная масштабируемость (особенно для однопроцессорных систем)
 - Не всегда эффективны для неструктурированных данных
-

Ключ-значение (например, Redis, Cassandra)

Преимущества:

- Очень высокая скорость чтения/записи
- Горизонтальное масштабирование
- Простота модели: ключ - значение
- Подходит для кэширования, сессий, быстрого доступа к данным по ключу

Недостатки:

- Нет поддержки SQL
 - Ограниченные возможности для сложных запросов (JOIN, GROUP BY)
 - Часто жертвуют согласованностью (CAP-теорема)
 - Нет транзакций между разными ключами
-

Hadoop / Spark (HDFS + MapReduce / Spark RDD)

Преимущества:

- Обработка огромных объёмов данных (Big Data)
- Высокая отказоустойчивость (HDFS реплицирует блоки)
- Поддержка неструктурированных данных (логи, JSON, CSV)

- Распределённая обработка (MapReduce, Spark)
- Spark позволяет писать SQL через Spark SQL / DataFrame API
- Подходит для тяжёлых ETL-задач

Недостатки:

- Высокая латентность — не подходит для оперативной аналитики
 - Сложность администрирования
 - Нет транзакций и ограничений (PK, FK)
 - Spark: нет INSERT/UPDATE/DELETE на уровне строк
-

Резидентные СУБД

(In-Memory DB, например, SAP HANA, Apache Ignite)

Преимущества:

- Очень высокая скорость — данные в оперативной памяти
- Поддержка аналитики в реальном времени
- Некоторые поддерживают SQL и транзакции
- Хороша для HTAP (Hybrid Transactional-Analytical Processing)

Недостатки:

- Ограничена объёмом RAM
 - Высокая стоимость хранения (RAM дороже диска)
 - Риск потери данных при сбое (если нет сохранения на диск)
 - Не подходит для очень больших исторических данных
-

Колоночные СУБД (ClickHouse и т.д.)

Преимущества:

- Очень высокая скорость агрегации и анализа
- Эффективное сжатие данных (по столбцам)
- Поддержка SQL (диалект)
- Параллельная и распределённая обработка
- Линейное масштабирование
- Оптимизирована для OLAP: много читаемых строк, мало столбцов

Недостатки:

- Плохо подходит для OLTP и точечных запросов (`SELECT * WHERE id = x`)

- Нет полноценных транзакций
 - Нельзя быстро обновлять или удалять строки
 - Разреженный индекс — неэффективен для поиска по вторичным полям
-