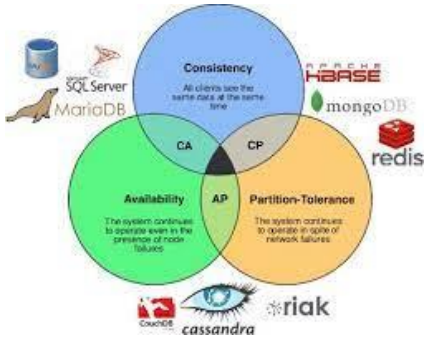
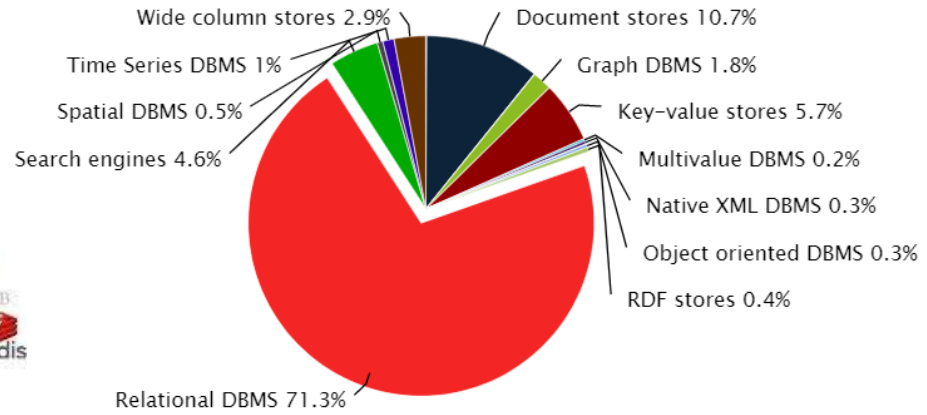
RDBMS vs NOSQL

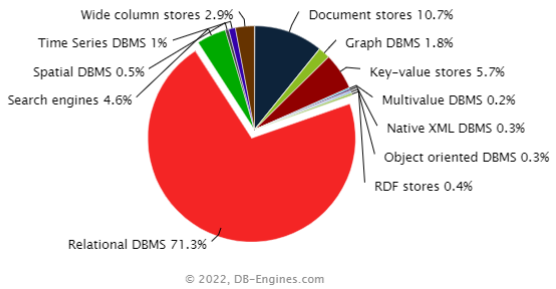




Mikail Saltan

dbengines.com rankings

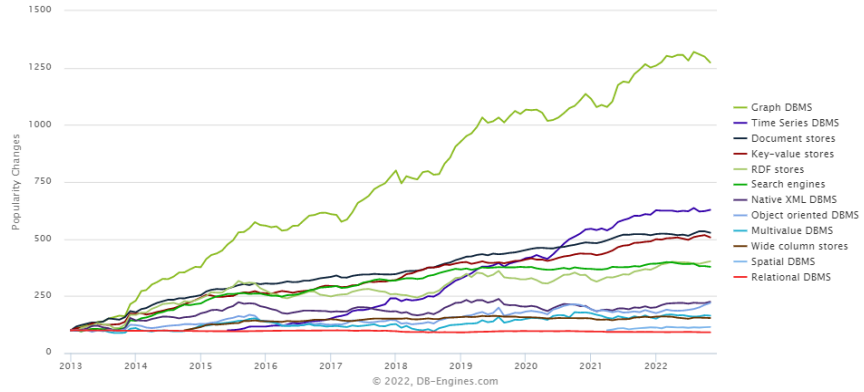
Ranking scores per category in percent, November 2022



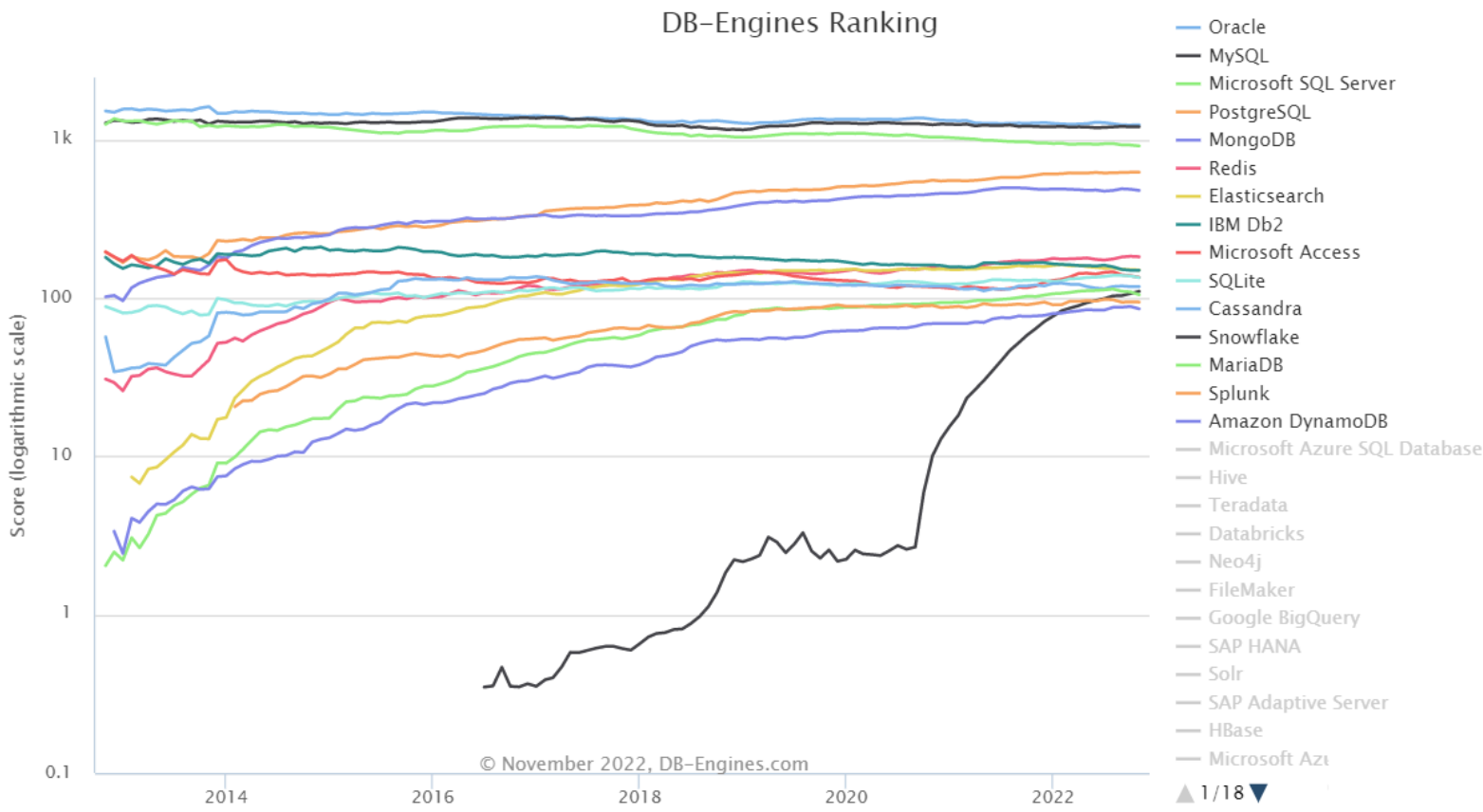
- Relational DBMS (71.3%)
- Document stores (10.7%)
- Key-value stores (5.7%)
- Search engines (4.6%)
- Wide column stores (2.9%)
- Graph DBMS (1.8%)
- Time Series DBMS (1.0 %)
- Others (2.0%)

Rank			DBMS	Database Model	Score		
Nov 2022	Oct 2022	Nov 2021			Nov 2022	Oct 2022	Nov 2021
1.	1.	1.	Oracle	Relational, Multi-model	1241.69	+5.32	-31.04
2.	2.	2.	MySQL	Relational, Multi-model	1205.54	+0.17	-5.98
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	912.51	-12.17	-41.78
4.	4.	4.	PostgreSQL	Relational, Multi-model	623.16	+0.44	+25.88
5.	5.	5.	MongoDB	Document, Multi-model	477.90	-8.33	-9.45
6.	6.	6.	Redis	Key-value, Multi-model	182.05	-1.33	+10.55
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	150.32	-0.74	-8.76
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	149.56	-0.10	-17.96
9.	9.	↑ 11.	Microsoft Access	Relational	135.03	-3.14	+15.79
10.	10.	↓ 9.	SQLite	Relational	134.63	-3.17	+4.83
11.	11.	↓ 10.	Cassandra	Wide column	118.12	+0.18	-2.76
12.	↑ 13.	↑ 18.	Snowflake	Relational	110.15	+3.43	+45.97
13.	↓ 12.	↓ 12.	MariaDB	Relational, Multi-model	104.91	-4.40	+2.72
14.	14.	↓ 13.	Splunk	Search engine	94.23	-0.43	+1.92
15.	15.	↑ 16.	Amazon DynamoDB	Multi-model	85.40	-2.95	+8.41
16.	16.	↓ 15.	Microsoft Azure SQL Database	Relational, Multi-model	83.66	-1.30	+2.34
17.	17.	↓ 14.	Hive	Relational	81.89	+1.29	-1.42
18.	18.	↓ 17.	Teradata	Relational, Multi-model	65.23	-0.84	-4.35
19.	↑ 20.		Databricks	Multi-model	60.89	+3.28	
20.	↓ 19.	↓ 19.	Neo4j	Graph	57.30	-1.38	-0.68
21.	↑ 23.	21.	FileMaker	Relational	54.31	+1.90	+0.08
22.	22.	↑ 24.	Google BigQuery	Relational	54.13	+1.68	+9.13
23.	↑ 24.	↓ 20.	SAP HANA	Relational, Multi-model	51.45	-0.63	-4.08
24.	↓ 21.	↓ 22.	Solr	Search engine, Multi-model	51.33	-2.17	-2.52
25.	25.	↓ 23.	SAP Adaptive Server	Relational, Multi-model	43.58	+0.62	-7.35
26.	26.	↓ 25.	HBase	Wide column	40.41	-1.25	-4.59
27.	27.	↓ 26.	Microsoft Azure Cosmos DB	Multi-model	39.75	-0.67	-1.08
28.	28.	↓ 27.	PostGIS	Spatial DBMS, Multi-model	30.78	-0.08	-1.14
29.	29.	29.	InfluxDB	Time Series, Multi-model	29.96	+0.38	+1.42
30.	30.	↓ 28.	Couchbase	Document, Multi-model	28.62	+0.66	-1.25

Complete trend, starting with January 2013



DB-Engines Ranking - Trend Popularity



RDBMS

☐ include secondary database models — 162 systems in ranking, November 2022

Rank Nov 2022	Rank Oct 2022	Rank Nov 2021	DBMS	Database Model	Score Nov 2022	Score Oct 2022	Score Nov 2021
1.	1.	1.	Oracle	Relational, Multi-model	1241.69	+5.32	-31.04
2.	2.	2.	MySQL	Relational, Multi-model	1205.54	+0.17	-5.98
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	912.51	-12.17	-41.78
4.	4.	4.	PostgreSQL	Relational, Multi-model	623.16	+0.44	+25.88
5.	5.	5.	IBM Db2	Relational, Multi-model	149.56	-0.10	-17.96
6.	6.	↑ 7.	Microsoft Access	Relational	135.03	-3.14	+15.79
7.	7.	↓ 6.	SQLite	Relational	134.63	-3.17	+4.83
8.	↑ 9.	↑ 12.	Snowflake	Relational	110.15	+3.43	+45.97
9.	↓ 8.	↓ 8.	MariaDB	Relational, Multi-model	104.91	-4.40	+2.72
10.	10.	10.	Microsoft Azure SQL Database	Relational, Multi-model	83.66	-1.30	+2.34
11.	11.	↓ 9.	Hive	Relational	81.89	+1.29	-1.42
12.	12.	↓ 11.	Teradata	Relational, Multi-model	65.23	-0.84	-4.35
13.	13.		Databricks	Multi-model	60.89	+3.28	
14.	↑ 15.	14.	FileMaker	Relational	54.31	+1.90	+0.08
15.	↓ 14.	↑ 16.	Google BigQuery	Relational	54.13	+1.68	+9.13
16.	16.	↓ 13.	SAP HANA	Relational, Multi-model	51.45	-0.63	-4.08
17.	17.	↓ 15.	SAP Adaptive Server	Relational, Multi-model	43.58	+0.62	-7.35
18.	18.	18.	Amazon Redshift	Relational	27.04	+0.03	+2.17
19.	19.	↓ 17.	Firebird	Relational	25.37	+0.36	-1.54
20.	20.	↑ 24.	Microsoft Azure Synapse Analytics	Relational	23.03	-0.07	+5.13
21.	21.	↓ 20.	Informix	Relational, Multi-model	22.82	-0.05	-0.41
22.	22.	↓ 19.	Spark SQL	Relational	21.90	-0.68	-1.43
23.	23.	↓ 21.	Vertica	Relational, Multi-model	19.70	-0.15	-0.63
24.	↑ 25.	↓ 23.	Impala	Relational, Multi-model	17.92	+0.24	-1.01
25.	↓ 24.	↓ 22.	Netezza	Relational	17.27	-0.57	-2.52
26.	↑ 27.	↑ 27.	Presto	Relational	15.15	+0.61	+1.67
27.	↓ 26.	↓ 25.	dBASE	Relational	14.25	-1.05	-0.52
28.	28.	↑ 29.	ClickHouse	Relational, Multi-model	13.88	+0.02	+2.49
29.	29.	↓ 28.	Amazon Aurora	Relational, Multi-model	12.66	-0.30	+0.56
30.	30.	↓ 26.	Greenplum	Relational, Multi-model	12.29	-0.21	-1.46
31.	31.	↓ 30.	H2	Relational, Multi-model	9.06	-0.51	-0.86
32.	32.	↓ 31.	Oracle Essbase	Relational	8.37	-0.40	-0.16
33.	33.	↓ 32.	CockroachDB	Relational	7.98	+0.03	+0.92
34.	↑ 35.	↑ 36.	SingleStore	Relational, Multi-model	7.81	+0.28	+1.47
35.	↓ 34.	↓ 34.	Microsoft Azure Data Explorer	Relational, Multi-model	7.34	-0.33	+0.72
36.	36.	↓ 35.	Ignite	Multi-model	6.14	-0.14	-0.23
37.	37.	↓ 33.	Interbase	Relational	6.04	-0.07	-0.62
38.	↑ 39.	↓ 37.	Ingres	Relational	5.93	+0.19	-0.10
39.	↑ 42.	↑ 41.	HyperSQL	Relational	5.91	+0.39	+0.85
40.	↓ 38.	↓ 38.	SAP SQL Anywhere	Relational	5.88	+0.05	+0.11
41.	↓ 40.	↑ 43.	Virtuoso	Multi-model	5.84	+0.12	+1.03
42.	↓ 41.	↓ 40.	Derby	Relational	5.66	+0.01	+0.55
43.	↑ 44.	↓ 42.	SAP IQ	Relational	4.72	+0.11	-0.26
44.	↑ 46.	↓ 39.	OpenEdge	Relational	4.50	+0.05	-0.64
45.	↓ 43.	45.	Oracle NoSQL	Multi-model	4.42	-0.25	+0.01

Document

include secondary database models — 55 systems in ranking, November 2022

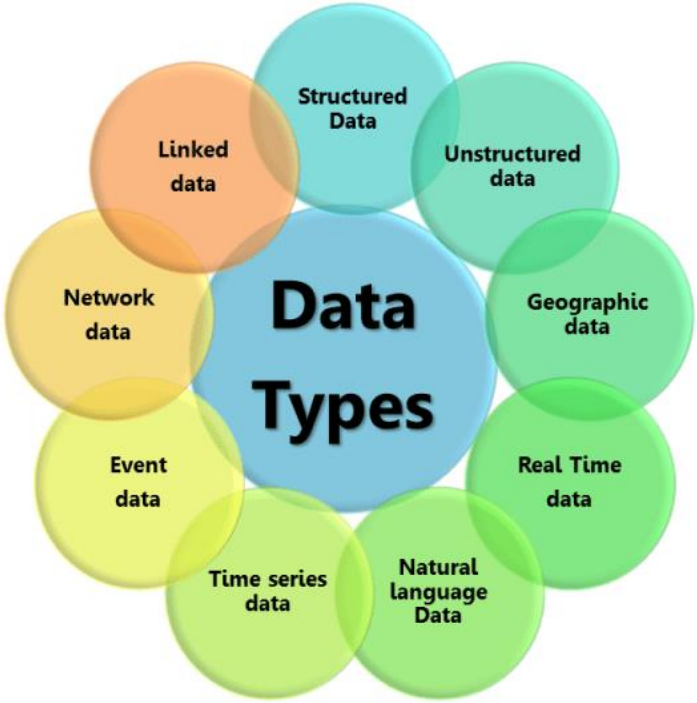
Rank Nov 2022	Rank Oct 2022	Rank Nov 2021	DBMS	Database Model	Score Nov 2022	Score Oct 2022	Score Nov 2021
1.	1.	1.	MongoDB	Document, Multi-model	477.90	-8.33	-9.45
2.	2.	2.	Amazon DynamoDB	Multi-model	85.40	-2.95	+8.41
3.	3.		Databricks	Multi-model	60.89	+3.28	
4.	4.	3.	Microsoft Azure Cosmos DB	Multi-model	39.75	-0.67	-1.08
5.	5.	4.	Couchbase	Document, Multi-model	28.62	+0.66	-1.25
6.	6.	5.	Firebase Realtime Database	Document	19.84	-0.29	-0.23
7.	7.	6.	CouchDB	Document, Multi-model	15.98	-0.27	-0.82
8.	8.	7.	Google Cloud Firestore	Document	11.09	-0.03	+1.53
9.	10.	8.	MarkLogic	Multi-model	9.75	+0.44	+0.41
10.	9.	9.	Realm	Document	9.03	-0.48	-0.17
11.	11.	12.	Google Cloud Datastore	Document	7.10	+0.17	+2.12
12.	12.	10.	Aerospike	Multi-model	6.95	+0.26	+1.51
13.	14.	13.	Virtuoso	Multi-model	5.84	+0.12	+1.03
14.	13.	11.	ArangoDB	Multi-model	5.84	-0.17	+0.74
15.	15.	14.	OrientDB	Multi-model	5.04	+0.06	+0.41
16.	17.	17.	RavenDB	Document, Multi-model	4.55	+0.14	+1.11
17.	16.	15.	Oracle NoSQL	Multi-model	4.42	-0.25	+0.01
18.	18.	16.	IBM Cloudant	Document	4.21	+0.35	-0.05
19.	19.	19.	RethinkDB	Document, Multi-model	3.48	-0.27	+0.29
20.	20.	18.	PouchDB	Document	3.44	-0.06	+0.10
21.	21.	20.	Apache Drill	Multi-model	3.33	+0.16	+0.36
22.	22.	21.	CloudKit	Document	2.87	0.00	+0.42
23.	23.	22.	InterSystems IRIS	Multi-model	2.52	+0.11	+0.47
24.	24.	23.	Amazon DocumentDB	Document	2.07	-0.14	+0.38
25.	26.	26.	FoundationDB	Multi-model	1.78	+0.12	+0.43
26.	25.	24.	Fauna	Multi-model	1.70	+0.00	+0.08
27.	27.	25.	Mnesia	Document	1.69	+0.07	+0.31
28.	28.	27.	LiteDB	Document	1.69	+0.10	+0.35
29.	29.	28.	Datameer	Document	1.54	+0.10	+0.26
30.	30.	29.	AllegroGraph	Multi-model	1.44	+0.06	+0.21
31.	31.	38.	GigaSpaces	Multi-model	1.30	+0.09	+0.78
32.	32.	30.	HPE Ezmeral Data Fabric	Multi-model	1.26	+0.13	+0.37
33.	33.	31.	BigchainDB	Document	1.07	+0.10	+0.26
34.	34.	35.	Rockset	Document, Multi-model	0.86	+0.10	+0.26
35.	35.	34.	AlaSQL	Multi-model	0.77	+0.02	+0.16
36.	36.	33.	Percona Server for MongoDB	Document	0.70	-0.04	+0.08
37.	38.	36.	LokiJS	Document	0.64	-0.03	+0.05
38.	40.	39.	Sequoiadb	Multi-model	0.60	+0.08	+0.14
39.	39.	32.	HarperDB	Document	0.59	+0.03	-0.14
40.	37.	37.	Yandex Database (YDB)	Multi-model	0.53	-0.18	-0.03
41.	41.	40.	EJDB	Document	0.39	+0.11	+0.15
42.	43.		Acebase	Document	0.28	+0.04	
43.	42.		SurrealDB	Multi-model	0.26	-0.01	
44.	44.	44.	Bangdb	Multi-model	0.20	-0.02	+0.16
45.	45.	41.	ArcadeDB	Multi-model	0.20	+0.01	+0.09

Key Value

☐ include secondary database models — 65 systems in ranking, November 2022

Rank Nov 2022	Rank Oct 2022	Rank Nov 2021	DBMS	Database Model	Score Nov 2022	Score Oct 2022	Score Nov 2021
1.	1.	1.	Redis	Key-value, Multi-model	182.05	-1.33	+10.55
2.	2.	2.	Amazon DynamoDB	Multi-model	85.40	-2.95	+8.41
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	39.75	-0.67	-1.08
4.	4.	4.	Memcached	Key-value	24.45	-0.95	-1.92
5.	5.	↑ 6.	Hazelcast	Key-value, Multi-model	9.60	-0.42	-0.40
6.	6.	↓ 5.	etcd	Key-value	8.71	-0.31	-2.10
7.	↑ 8.	↑ 10.	Aerospike	Multi-model	6.95	+0.26	+1.51
8.	↓ 7.	8.	Ehcache	Key-value	6.66	-0.16	+0.04
9.	9.	9.	Ignite	Multi-model	6.14	-0.14	-0.23
10.	10.	↓ 7.	Riak KV	Key-value	6.09	-0.18	-0.63
11.	11.	11.	ArangoDB	Multi-model	5.84	-0.17	+0.74
12.	12.	↑ 14.	Google Cloud Bigtable	Multi-model	5.24	-0.20	+0.97
13.	13.	↓ 12.	OrientDB	Multi-model	5.04	+0.06	+0.41
14.	↑ 15.	↑ 15.	RocksDB	Key-value	4.61	+0.11	+0.77
15.	↓ 14.	↓ 13.	Oracle NoSQL	Multi-model	4.42	-0.25	+0.01
16.	16.	↑ 19.	LevelDB	Key-value	3.68	+0.00	+0.63
17.	17.	↑ 18.	Infinispan	Key-value	3.42	-0.01	+0.34
18.	18.	↓ 17.	Oracle Berkeley DB	Multi-model	3.26	+0.06	-0.01
19.	19.	↓ 16.	InterSystems Caché	Multi-model	2.94	-0.05	-0.36
20.	20.	20.	Oracle Coherence	Key-value	2.69	+0.11	-0.32
21.	21.	↑ 22.	Amazon SimpleDB	Key-value	2.65	+0.08	+0.39
22.	22.	↓ 21.	LMDB	Key-value	2.60	+0.17	-0.17
23.	23.	23.	InterSystems IRIS	Multi-model	2.52	+0.11	+0.47
24.	24.	24.	GridGain	Multi-model	2.31	-0.03	+0.38
25.	25.	↑ 26.	Geode	Key-value	2.25	+0.18	+0.64
26.	26.	↓ 25.	Tarantool	Key-value, Multi-model	1.92	+0.16	+0.28
27.	27.	↑ 28.	FoundationDB	Multi-model	1.78	+0.12	+0.43
28.	28.	↓ 27.	GT.M	Key-value	1.49	-0.05	+0.07
29.	↑ 31.	↑ 30.	ZODB	Key-value	1.31	+0.10	+0.35
30.	↓ 29.	↓ 29.	NCache	Key-value, Multi-model	1.30	+0.08	+0.26
31.	↓ 30.	31.	Graph Engine	Multi-model	1.27	+0.06	+0.44
32.	↑ 33.	↑ 38.	Hibari	Key-value	0.98	+0.18	+0.55
33.	↑ 34.	↓ 32.	WebSphere eXtreme Scale	Key-value	0.86	+0.06	+0.24
34.	↓ 32.	↓ 33.	BoltDB	Key-value	0.85	+0.04	+0.29
35.	35.	35.	MapDB	Key-value	0.84	+0.11	+0.30
36.	↑ 38.	↑ 37.	Project Voldemort	Key-value	0.62	+0.07	+0.18
37.	↑ 39.	↓ 34.	Tokyo Tyrant	Key-value	0.59	+0.05	+0.03
38.	↓ 37.	↑ 39.	Scalaris	Key-value	0.57	+0.01	+0.14
39.	↓ 36.	↑ 40.	KeyDB	Key-value	0.56	-0.02	+0.22
40.	40.	↑ 46.	Cloudflare Workers KV	Key-value	0.48	+0.07	+0.40
41.	41.	41.	LeanXcale	Multi-model	0.26	-0.04	+0.03
42.	↑ 44.		Speedb	Key-value	0.26	+0.04	
43.	↑ 49.	↑ 49.	Elliptics	Key-value	0.25	+0.11	+0.19
44.	↓ 42.	↑ 52.	Immudb	Key-value, Multi-model	0.22	-0.02	+0.21
45.	45.	45.	ArcadeDB	Multi-model	0.20	+0.01	+0.09

Data Types



Unstructured Data

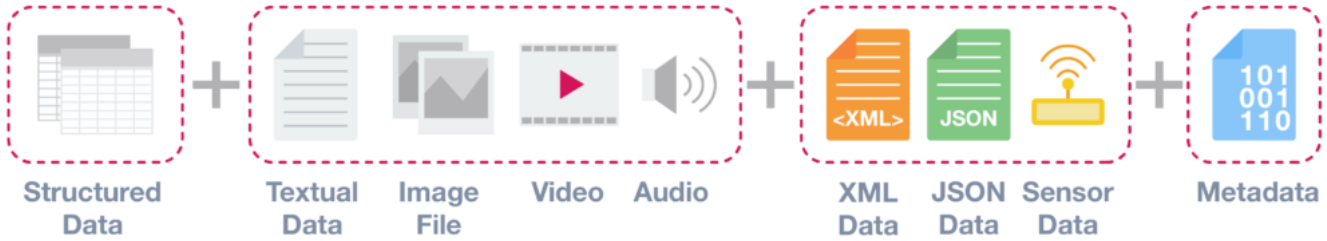
The university has 5600 students. Shaun (ID Number: 160801), 18 years old Communication study. Linh with ID number 160802, majoring in Accounting and is 20 years old. Ahmed from Psychology study program, 19 years old, ID number 160803.

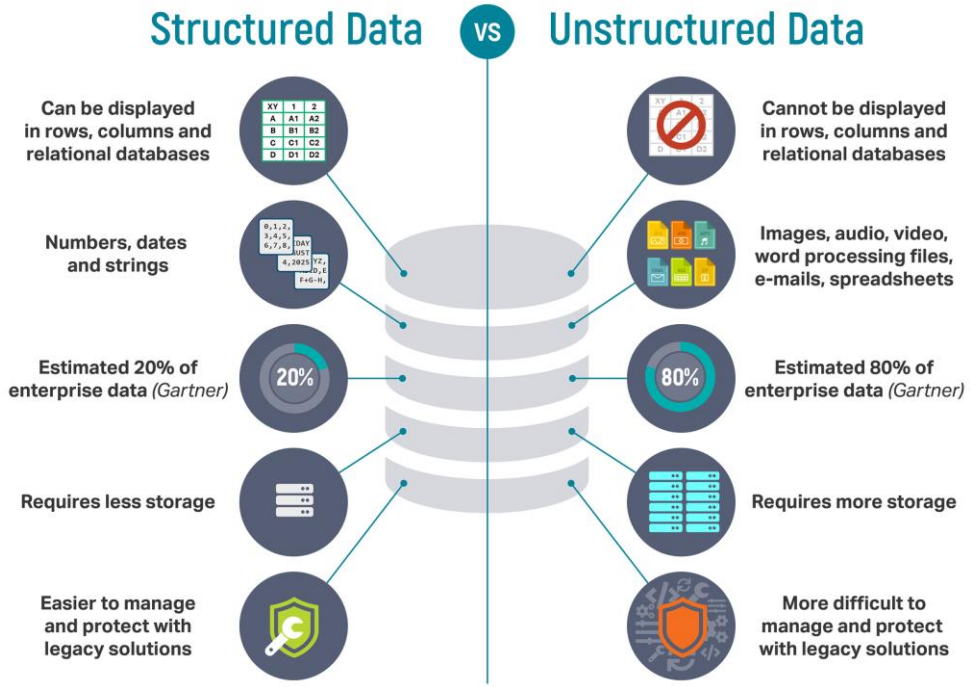
Semi-Structured Data

```
<University>
 <ID Number="160801">
  <Name="Shaun">
  <Age="18">
  <Program="Communication">
 <ID Number="160802">
  <Name="Linh">
  <Age="20">
  <Program="Accounting">
..... </University>
```

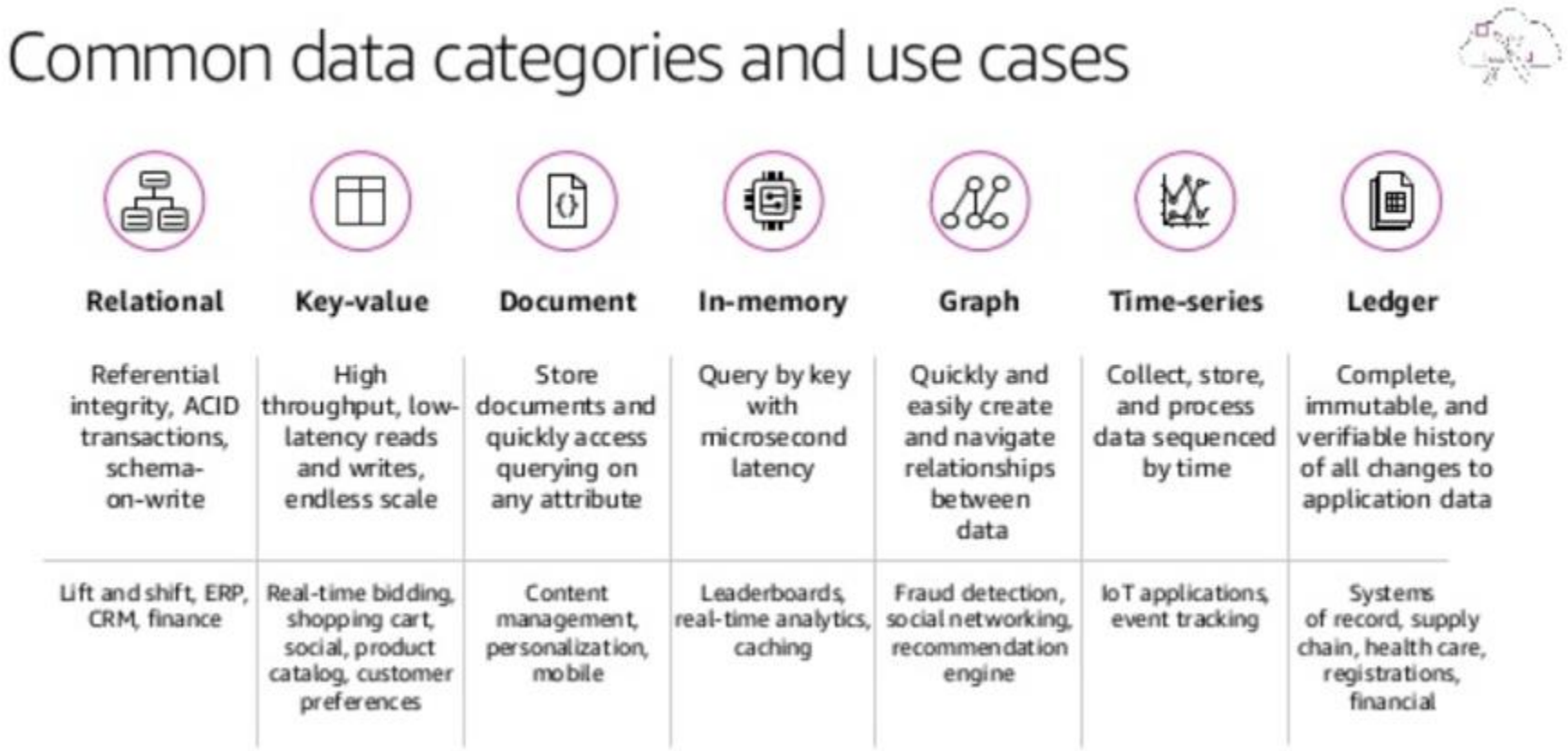
Structured Data

ID	Name	Age	Program
160801	Shaun	18	Communication
160802	Linh	20	Accounting
160803	Ahmed	19	Psychology

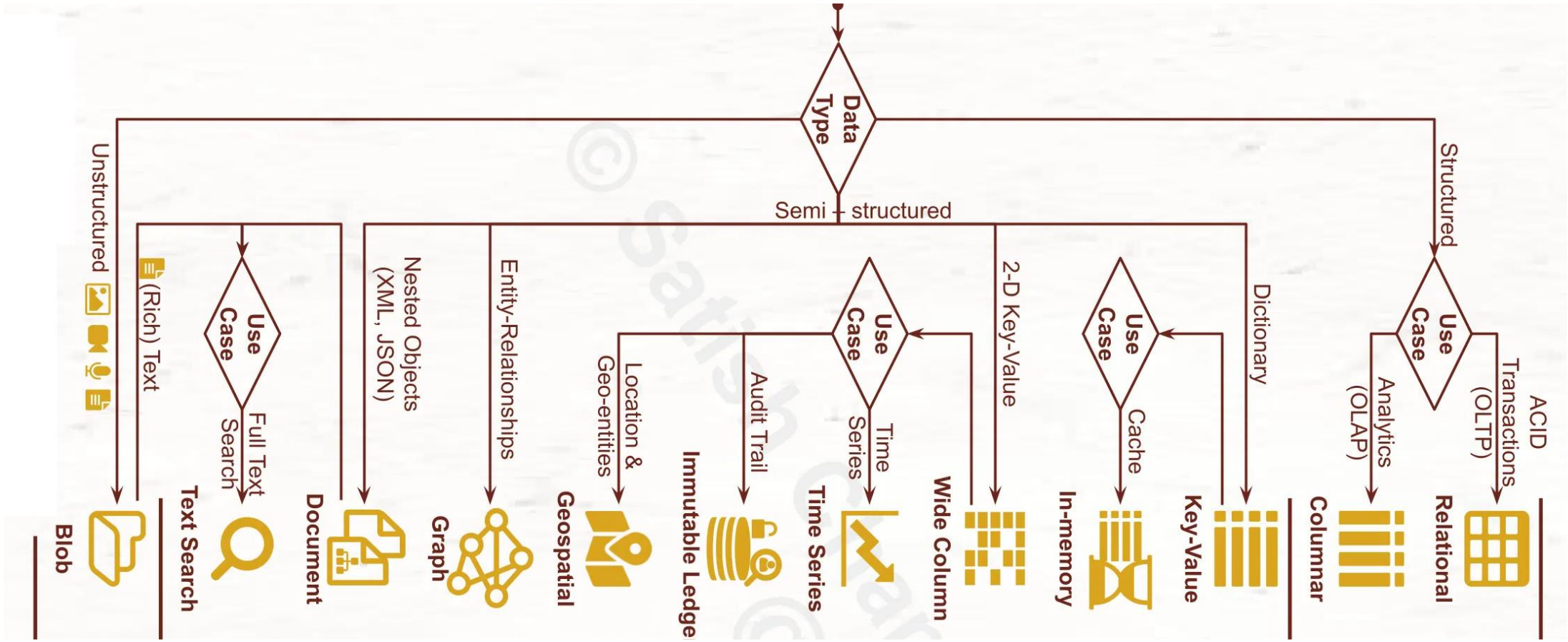




Data Types – Common Categories by Amazon - How to Choose the Right Database



Data Types – Common Categories - How to Choose the Right Database



NOSQL Database Types

Type	Notable examples of this type
Key–value cache	Apache Ignite, Couchbase, Coherence, eXtreme Scale, Hazelcast, Infinispan, Memcached, Redis, Velocity
Key–value store	Azure Cosmos DB, ArangoDB, Amazon DynamoDB, Aerospike, Couchbase, ScyllaDB
Key–value store (eventually consistent)	Azure Cosmos DB, Oracle NoSQL Database, Riak, Voldemort
Key–value store (ordered)	FoundationDB, InfinityDB, LMDB, MemcacheDB
Tuple store	Apache River, GigaSpaces, Tarantool, TIBCO ActiveSpaces, OpenLink Virtuoso
Triplestore	AllegroGraph, MarkLogic, Ontotext-OWLIM, Oracle NoSQL database, Profium Sense, Virtuoso Universal Server
Object database	Objectivity/DB, Perst, ZopeDB, db4o, GemStone/S, InterSystems Caché, JADE, ObjectDatabase++, ObjectDB, ObjectStore, ODABA, Realm, OpenLink Virtuoso, Versant Object Database, ZODB
Document store	Azure Cosmos DB, ArangoDB, BaseX, Clusterpoint, Couchbase, CouchDB, DocumentDB, eXist-db, IBM Domino, MarkLogic, MongoDB, RavenDB, Qizx, RethinkDB, Elasticsearch, OrientDB
Wide Column Store	Azure Cosmos DB, Amazon DynamoDB, Bigtable, Cassandra, Google Cloud Datastore, HBase, Hypertable, ScyllaDB
Native multi-model database	ArangoDB, Azure Cosmos DB, OrientDB, MarkLogic, Apache Ignite,[22][23] Couchbase, FoundationDB, Oracle Database
Graph database	Azure Cosmos DB, AllegroGraph, ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso
Multivalue database	D3 Pick database, Extensible Storage Engine (ESE/NT), InfinityDB, InterSystems Caché, jBASE Pick database, mvBase Rocket Software, mvEnterprise Rocket Software, Northgate Information Solutions Reality (the original Pick/MV Database), OpenQM, Revelation Software's OpenInsight (Windows) and Advanced Revelation (DOS), UniData Rocket U2, UniVerse Rocket U2

A-Structured Data

Normalization: 1NF, 2NF, 3NF

First normal form (1NF) – entity whose attributes have no more than one value for a single instance of that entity

- Any attributes that can have multiple values actually describe a separate entity, possibly an entity and relationship.

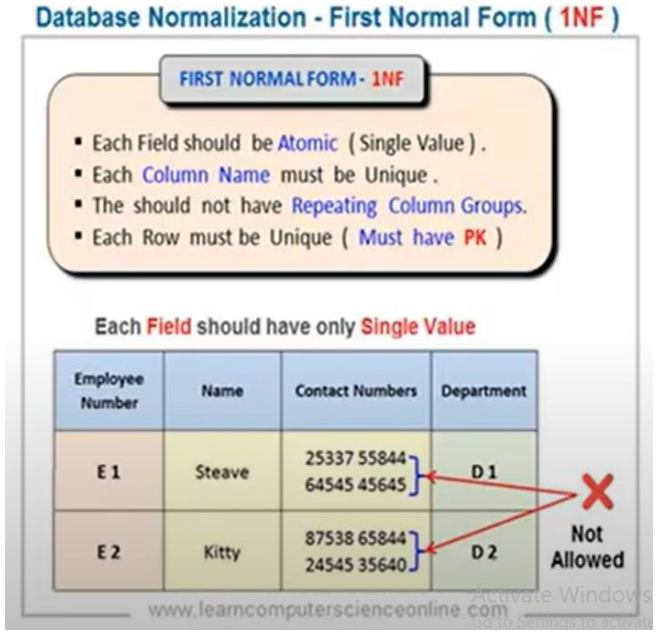
Second normal form (2NF) – entity whose nonprimary-key attributes are dependent on the full primary key.

- Any nonkey attributes dependent on only part of the primary key should be moved to entity where that partial key is the full key. May require creating a new entity and relationship on the model.

Third normal form (3NF) – entity whose nonprimary-key attributes are not dependent on any other non-primary key attributes.

- Any nonkey attributes that are dependent on other nonkey attributes must be moved or deleted. Again, new entities and relationships may have to be added to the data model.

8-1



A-Structured Data

Normalization: 1NF, 2NF, 3NF

First normal form (1NF) – entity whose attributes have no more than one value for a single instance of that entity

- Any attributes that can have multiple values actually describe a separate entity, possibly an entity and relationship.

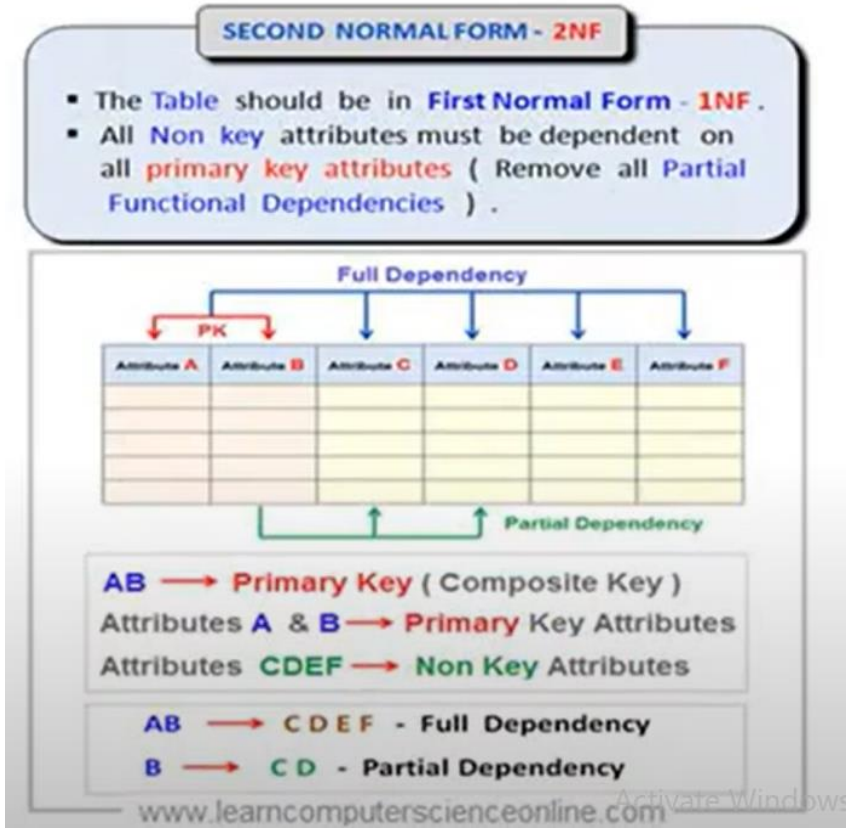
Second normal form (2NF) – entity whose nonprimary-key attributes are dependent on the full primary key.

- Any nonkey attributes dependent on only part of the primary key should be moved to entity where that partial key is the full key. May require creating a new entity and relationship on the model.

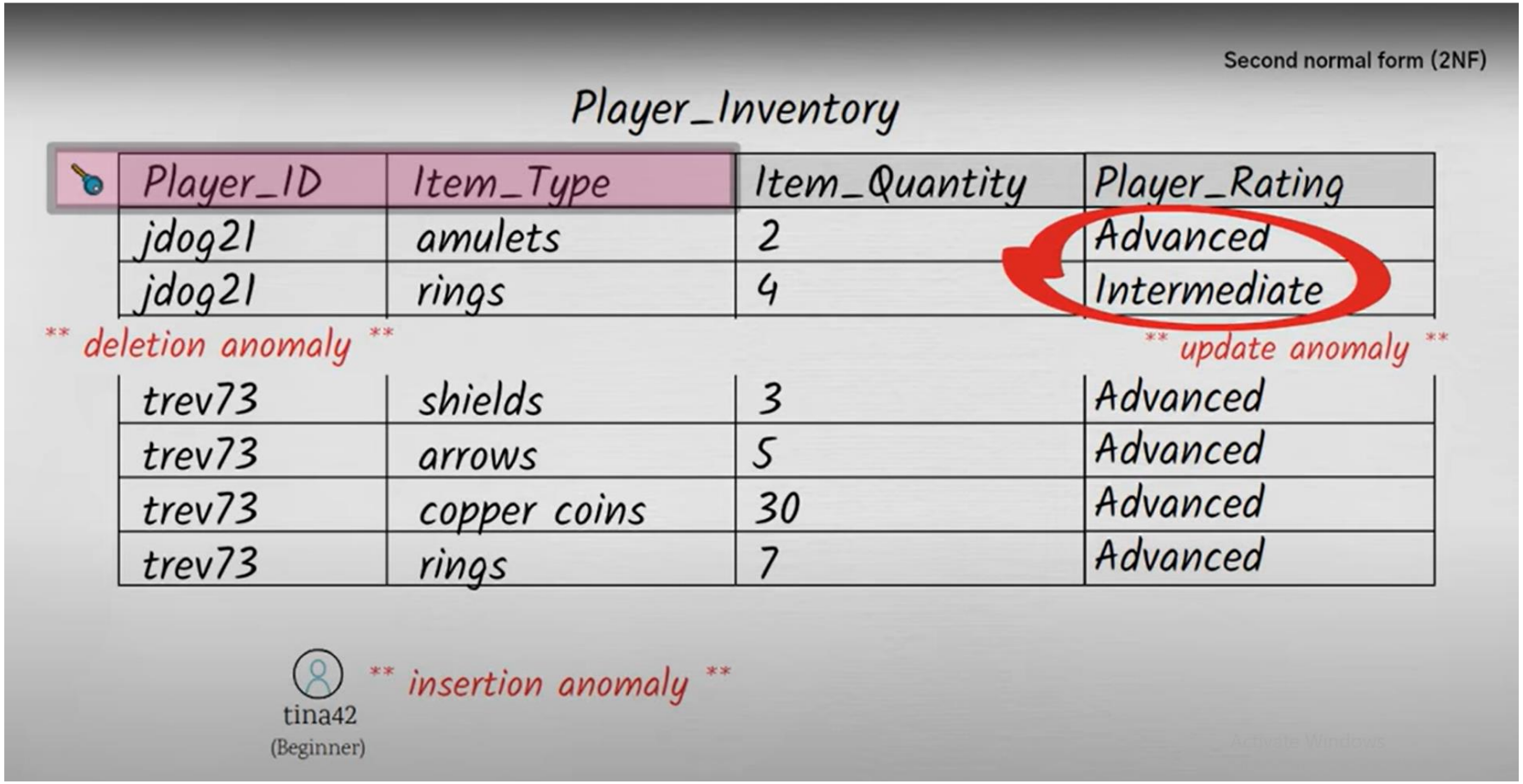
Third normal form (3NF) – entity whose nonprimary-key attributes are not dependent on any other non-primary key attributes.

- Any nonkey attributes that are dependent on other nonkey attributes must be moved or deleted. Again, new entities and relationships may have to be added to the data model.

8-1



A-Structured Data-Without 2NF



A-Structured Data

Normalization: 1NF, 2NF, 3NF

First normal form (1NF) – entity whose attributes have no more than one value for a single instance of that entity

- Any attributes that can have multiple values actually describe a separate entity, possibly an entity and relationship.

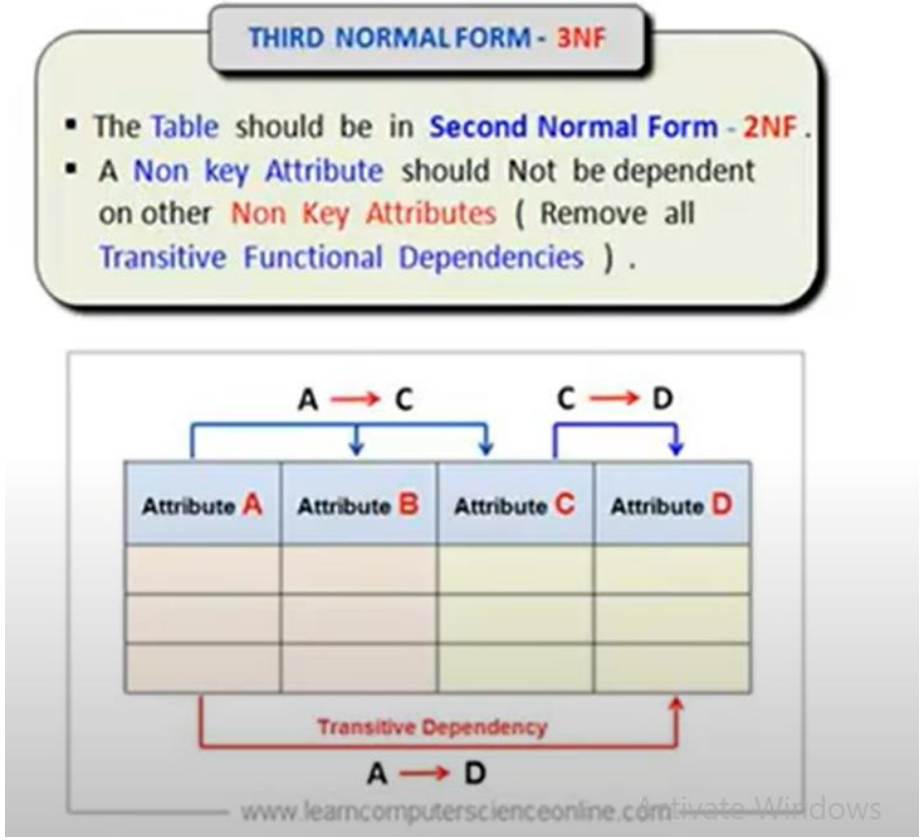
Second normal form (2NF) – entity whose nonprimary-key attributes are dependent on the full primary key.

- Any nonkey attributes dependent on only part of the primary key should be moved to entity where that partial key is the full key. May require creating a new entity and relationship on the model.

Third normal form (3NF) – entity whose nonprimary-key attributes are not dependent on any other non-primary key attributes.

- Any nonkey attributes that are dependent on other nonkey attributes must be moved or deleted. Again, new entities and relationships may have to be added to the data model.

8-1



1NF

Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

2NF

Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Membership ID	Full names	Physical Address	Salutation
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

3NF

Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Membership ID	Full Names	Physical Address	Salutation
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Membership ID	Movies Rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Membership ID	Full Names	Physical Address	Salutation ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1

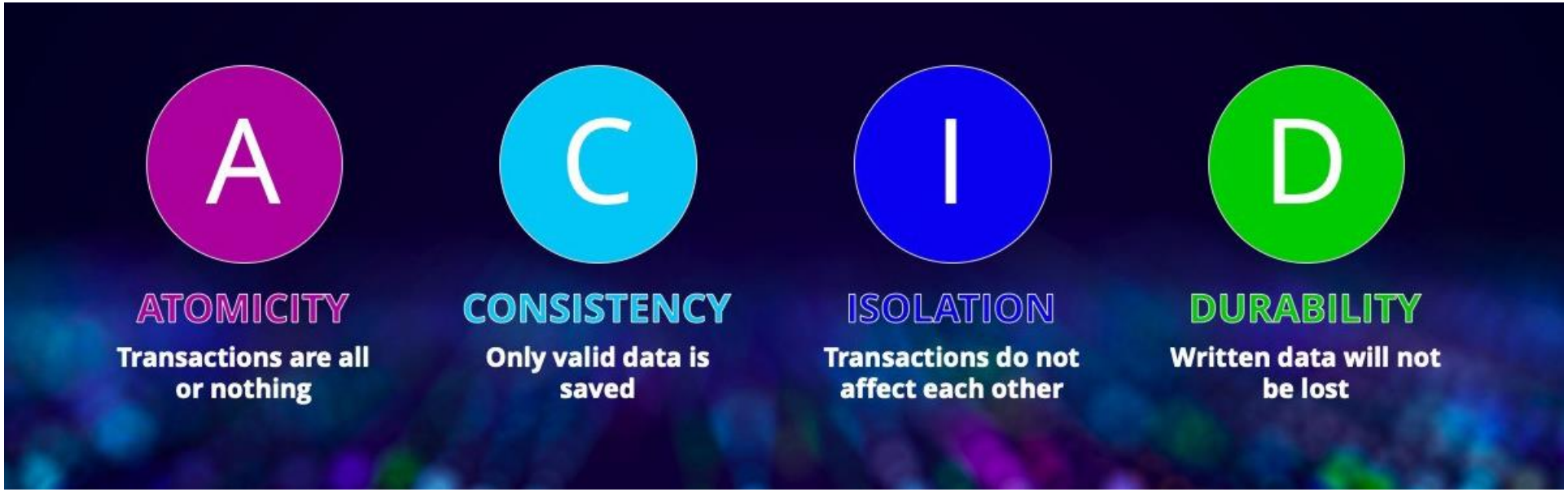
Salutation ID	Salutation
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

Membership ID	Movies Rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

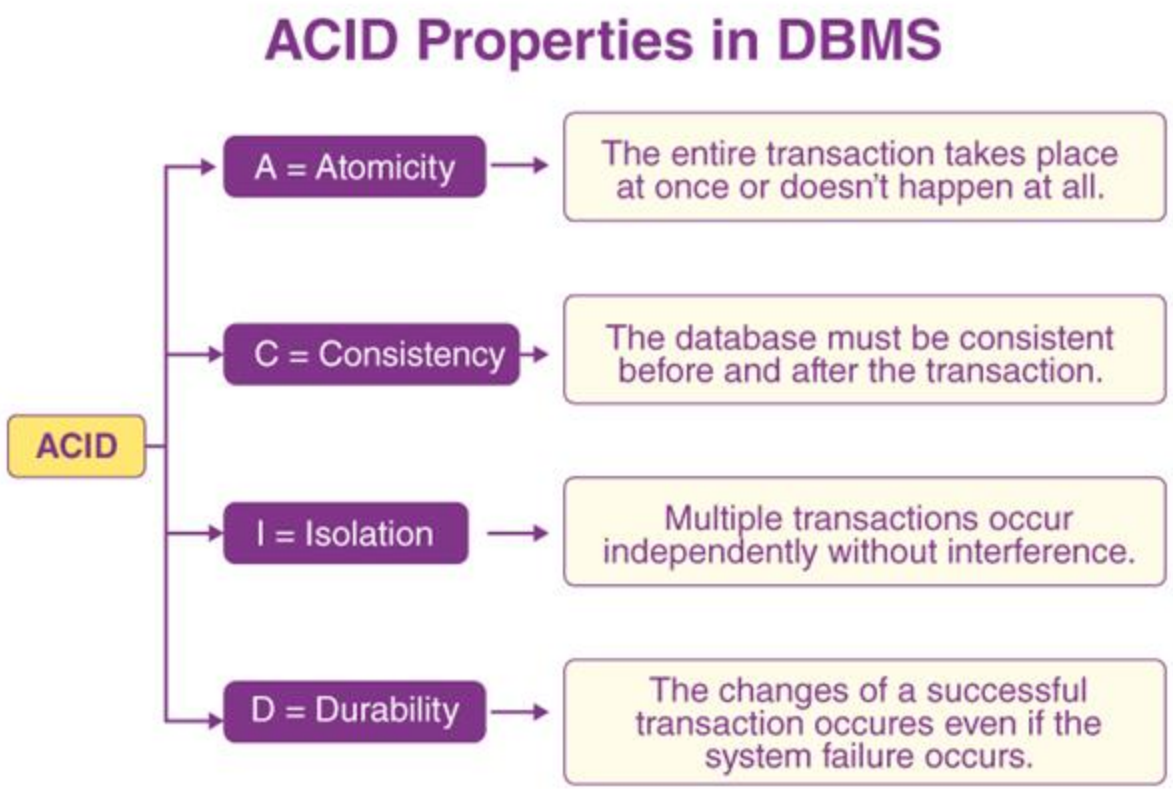
Bad Practices in Database Design: Are You Making These Mistakes?

- 1. Poor Design/Planning
- 2. Ignoring Normalization
- 3. Redundant Records
- 4. Poor Naming Standards
- 5. Lack of Documentation
- 6. One Table to Hold All Domain Values
- 7. Ignoring Frequency or Purpose of the Data
- 8. Insufficient Indexing
- 9. Not Taking Advantage of Database Features
- 10. Lack of Testing

A-Structured Data



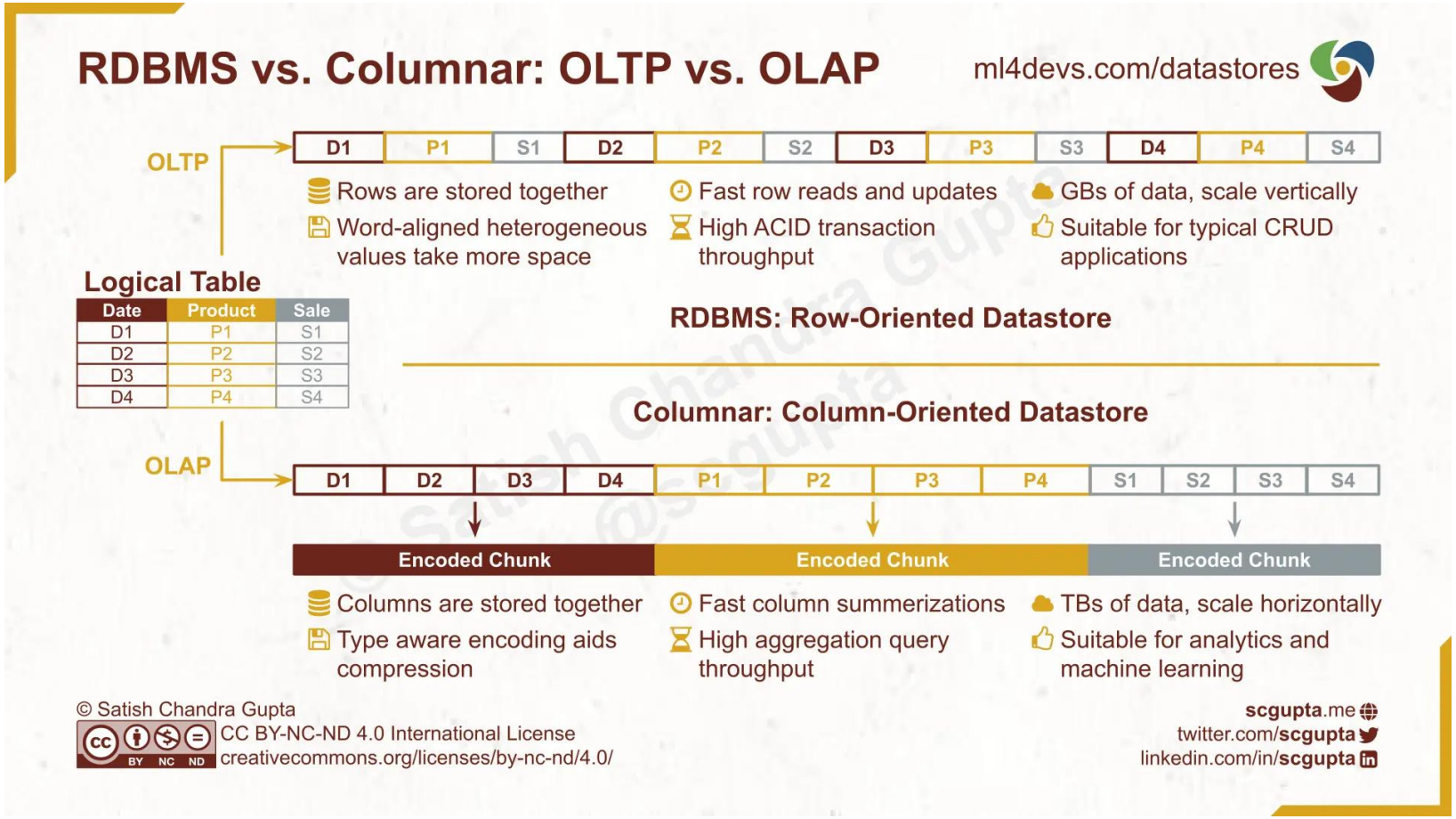
ACID Consideration



NOSQL ACID

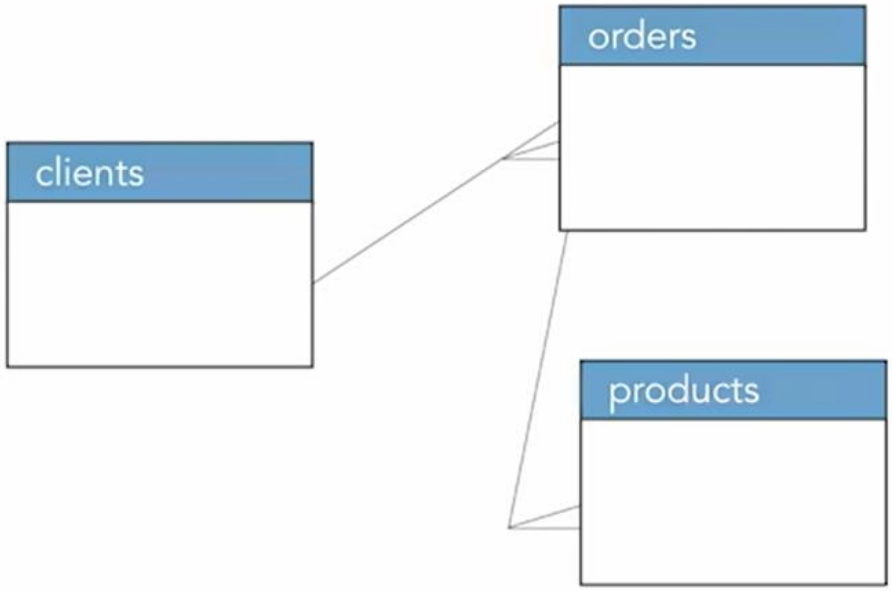
Database	ACID	Joins
Aerospike	Yes	No
Apache Ignite	Yes	Yes
ArangoDB	Yes	Yes
Amazon DynamoDB	Yes	No
Couchbase	Yes	Yes
CouchDB	Yes	Yes
IBM Db2	Yes	Yes
InfinityDB	Yes	No
LMDB	Yes	No
MarkLogic	Yes	Yes[nb 1]
MongoDB	Yes	Yes[nb 2]
OrientDB	Yes	Yes[nb 3]

A-Structured Data



Structured – Relational- RDBMS Assessment

- Designed for fixed schemas
- Highly structured rows and tables
- Query with SQL
- ACID transactions
- For example, PostgreSQL

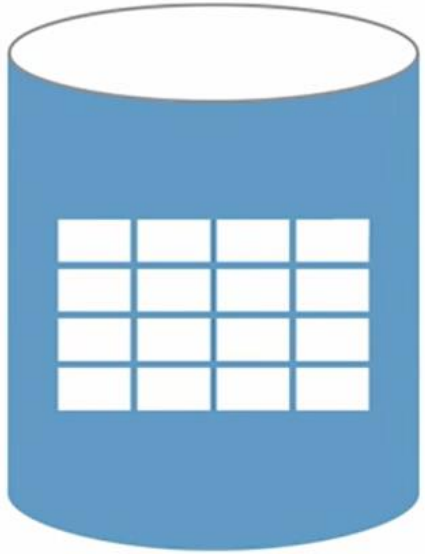


RDBMS Assessment – Oracle, PostgreSQL, MsSQL

Strengths	Shortcomings	Applications
• Transactions • Speed • Data integrity • Security	• Linear scaling • Text/media • Complex data types • Cost	• Master data • OLTP • Update intensive data • Data warehouse

Structured - Columnar - Columnar DB Assessment

- Semistructured
- Sparse multidimensional matrix
- Related data stored together in single row
- May use column families
- For example, Cassandra



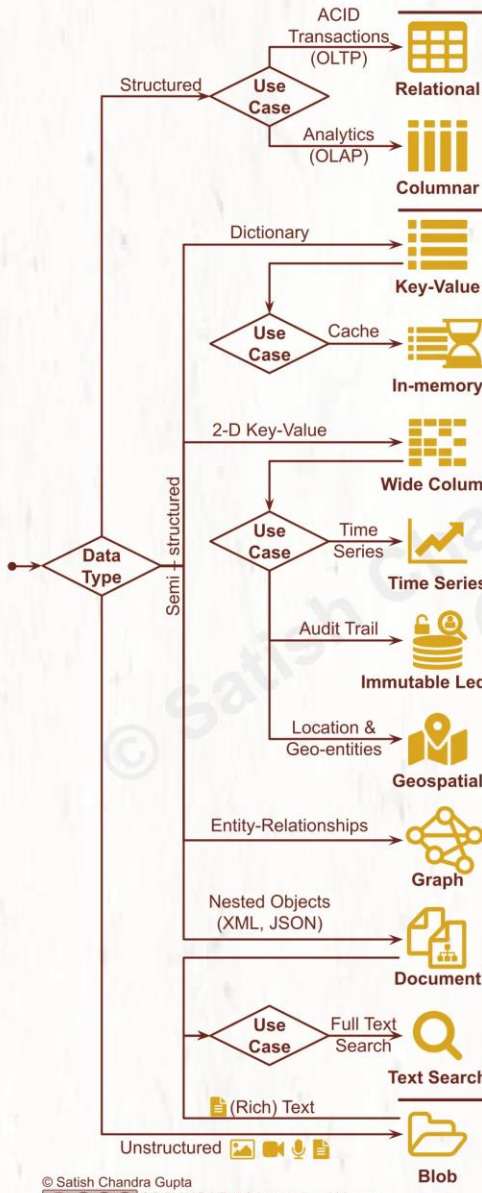
Wide-Column Databases

Columnar Assessment - Cassandra

Strengths	Shortcomings	Applications
• CQL – like SQL • Optimal updates • Transactions • Scalability	• No joins • Query by key only • No order by • Only small blobs	• Data warehouse • Real-time counters • Customer 360 • Write intensive apps • Machine learning

SQL vs. NoSQL: Cheatsheet for AWS, Azure, and Google Cloud

ml4devs.com/datastores



	aws	Azure	Google Cloud	Cloud Agnostic
Relational	RDS, Aurora	Azure SQL Database	Cloud SQL, Cloud Spanner	SQL Server, Oracle, DB2, MySQL, PostgreSQL
Columnar	RedShift	Azure Synapse	BigQuery	Snowflake, ClickHouse, Druid, Pinot, Databricks
Key-Value	DynamoDB	Cosmos DB	BigTable	Redis, ScyllaDB, Ignite
In-memory	ElastiCache	Azure Cache for Redis	Memory-store	Redis, Memcached, Hazelcast, Ignite
Wide Column	Keyspaces	Cosmos DB	BigTable	HBase, Cassandra, ScyllaDB
Time Series	Timestream	Cosmos DB	BigTable, BigQuery	TimescaleDB, OpenTSDB, InfluxDB, ScyllaDB
Immutable Ledger	Quantum Ledger Database (QLDB)	Azure SQL Database Ledger	✖	Hyperledger Fabric
Geospatial	Keyspaces	Cosmos DB	BigTable, BigQuery	Solr, PostGIS, MongoDB (GeoJSON)
Graph	Neptune	Cosmos DB	JanusGraph + BigTable	OrientDB, Neo4J, Giraph
Document	Document DB	Cosmos DB	Firestore	MongoDB, Couchbase, Solr
Text Search	Open-Search, Cloud-Search	Cognitive Search	Search APIs on Datastores	Elastic-Search, Solr, Elassandra
Blob	S3	Blob Storage	Cloud Storage	HDFS, MinIO

NoSQL

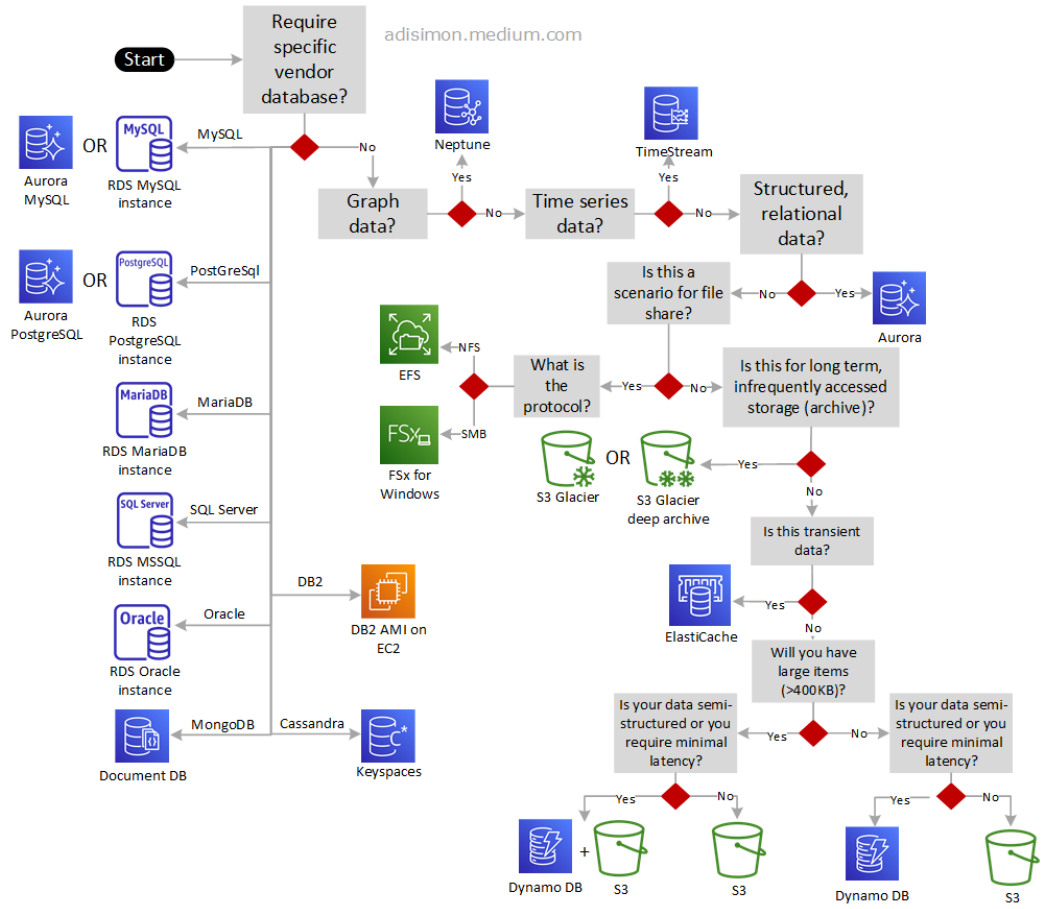
scgupta.me
twitter.com/scgupta
linkedin.com/in/scgupta

Comparison of NoSQL databases

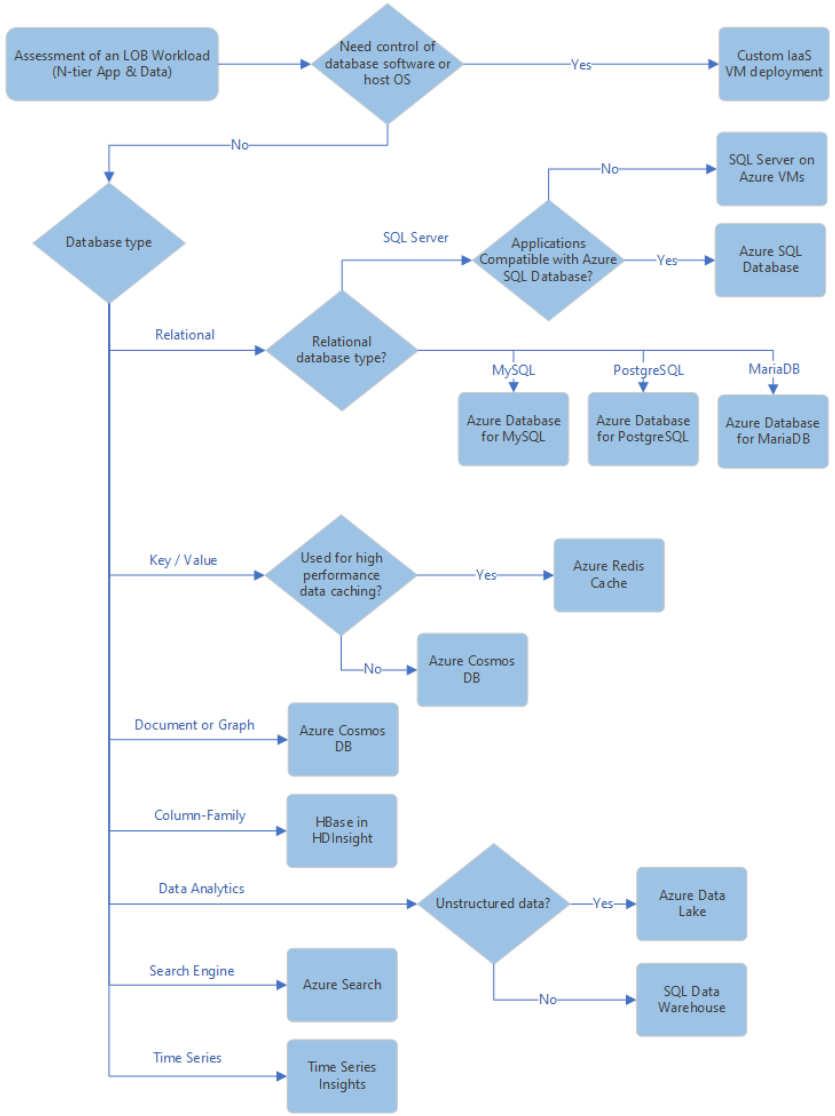
DATABASE	TYPE	VENDOR OR OPEN SOURCE	ACID COMPLIANCE	PRIMARY QUERY LANGUAGE	TOP USE CASES	SECURITY
Couchbase	Document-based, key value	Open source	Yes	N1QL	Customer service, financial services, inventory and IoT	Includes security for authentication, encryption, auditing and authorization
Cassandra	Wide column	Open source	No	CQL	Social analytics, real-time analytics, retail and messaging	Built-in security for authorization, encryption and authentication, but security is disabled by default for ease of use within clusters
Neo4j	Graph	Open source single-node version; commercial license for clustering	Yes	Cypher	AI, master data management, recommendation services and fraud protection	Built-in security for authorization, roles and encryption
Google Cloud Bigtable	Wide column	Vendor	No	Allows for use of many languages	IoT data management, financial services, retail data and time series data	Secured by vendor
Redis	Key value	Open source	Yes	Allows for use of many languages	Caching, queuing, filtering and stats	Automatically starts in "protection mode" and offers security suggestions
MongoDB	Document-based	Limited open source version; advanced features require commercial subscription	Yes	JavaScript	IoT management, real-time analytics, app development, inventory and personalization	Built-in security for authorization, authentication and encryption
Amazon DynamoDB	Key value or document-based	Vendor	Yes	DQL	Gaming, retail, financial services, advertising and streaming media	Built-in security for data and applications; vendor-secured software, hardware, facilities and network

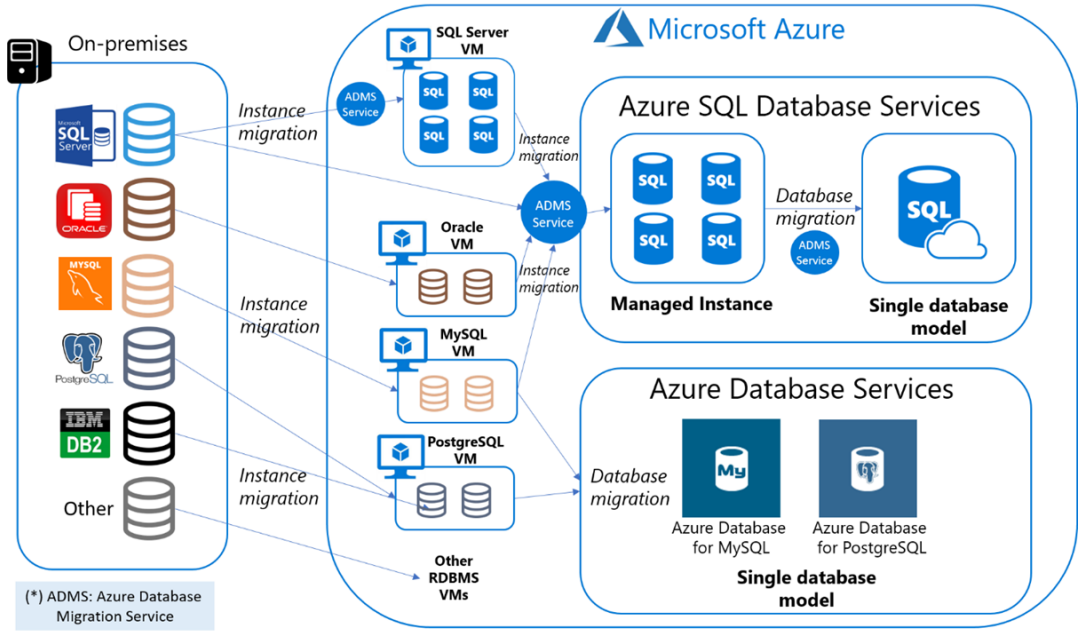
*BEFORE DOWNLOADING DATABASE, CHECK FOR SPECIFIC RECOMMENDATIONS TO SUPPORT YOUR DATA SETS.

Amozon data decision tree

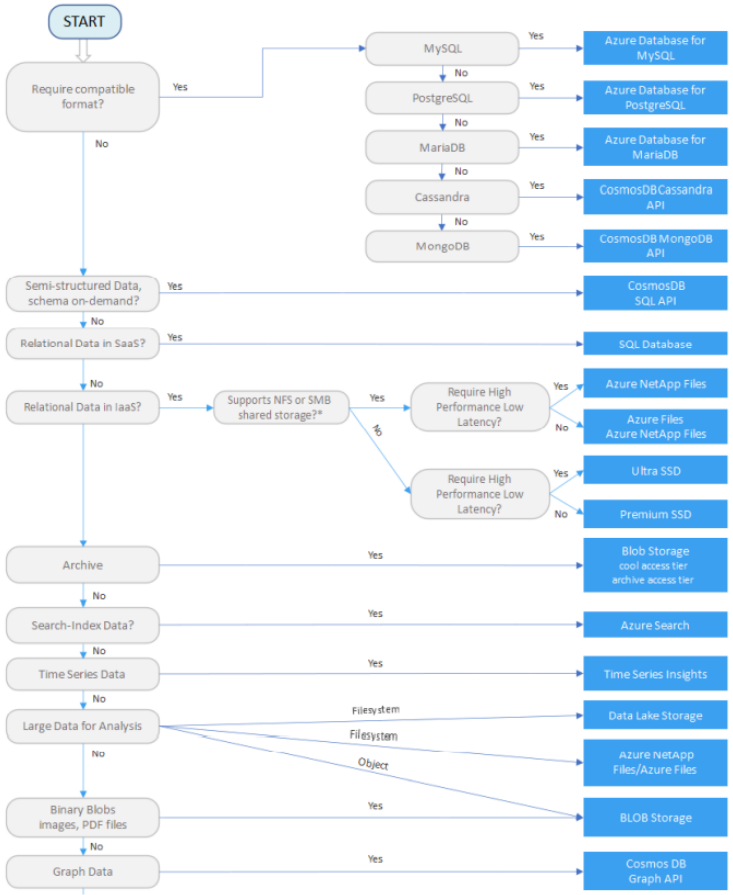


Azure data decision tree

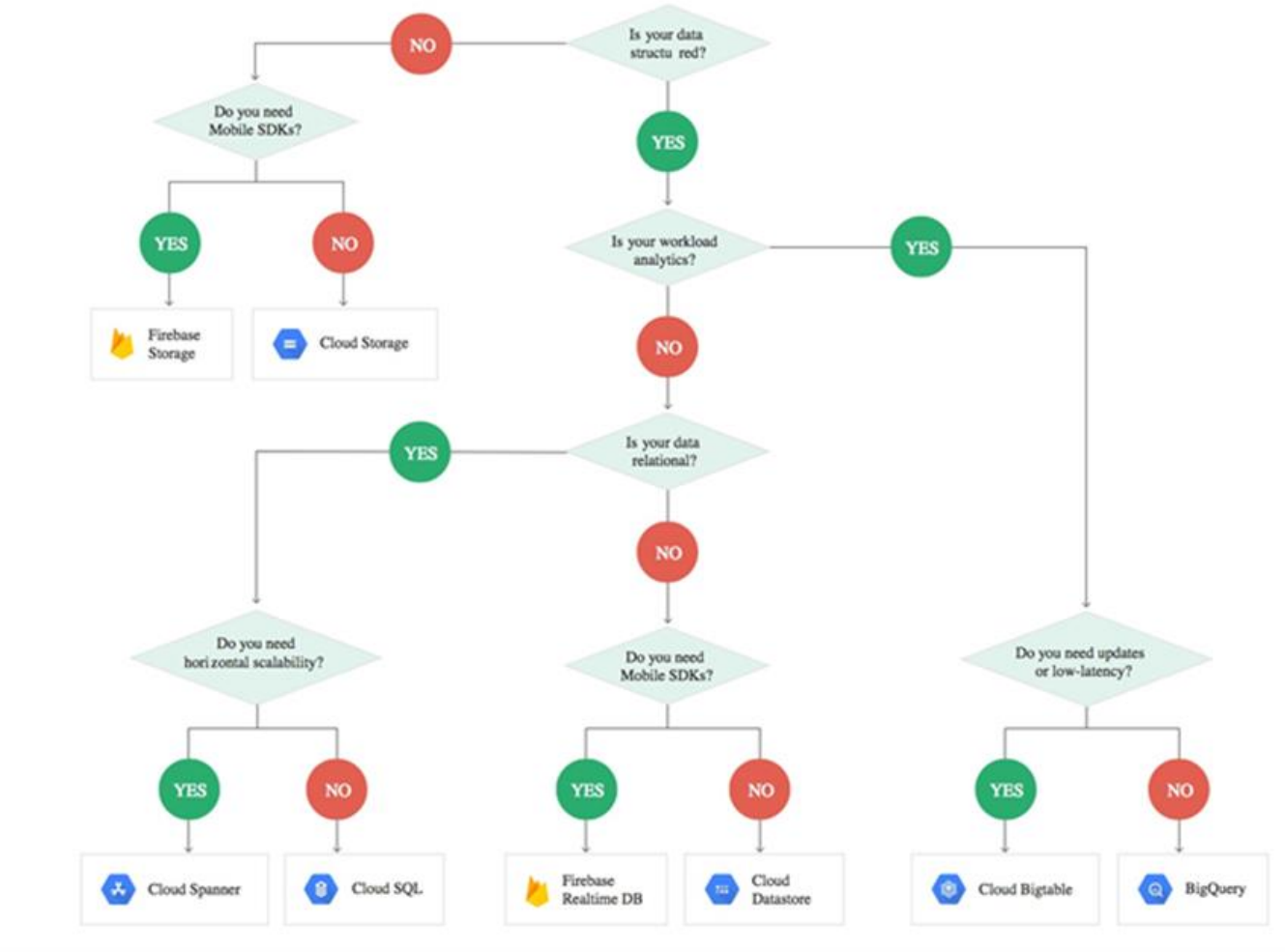




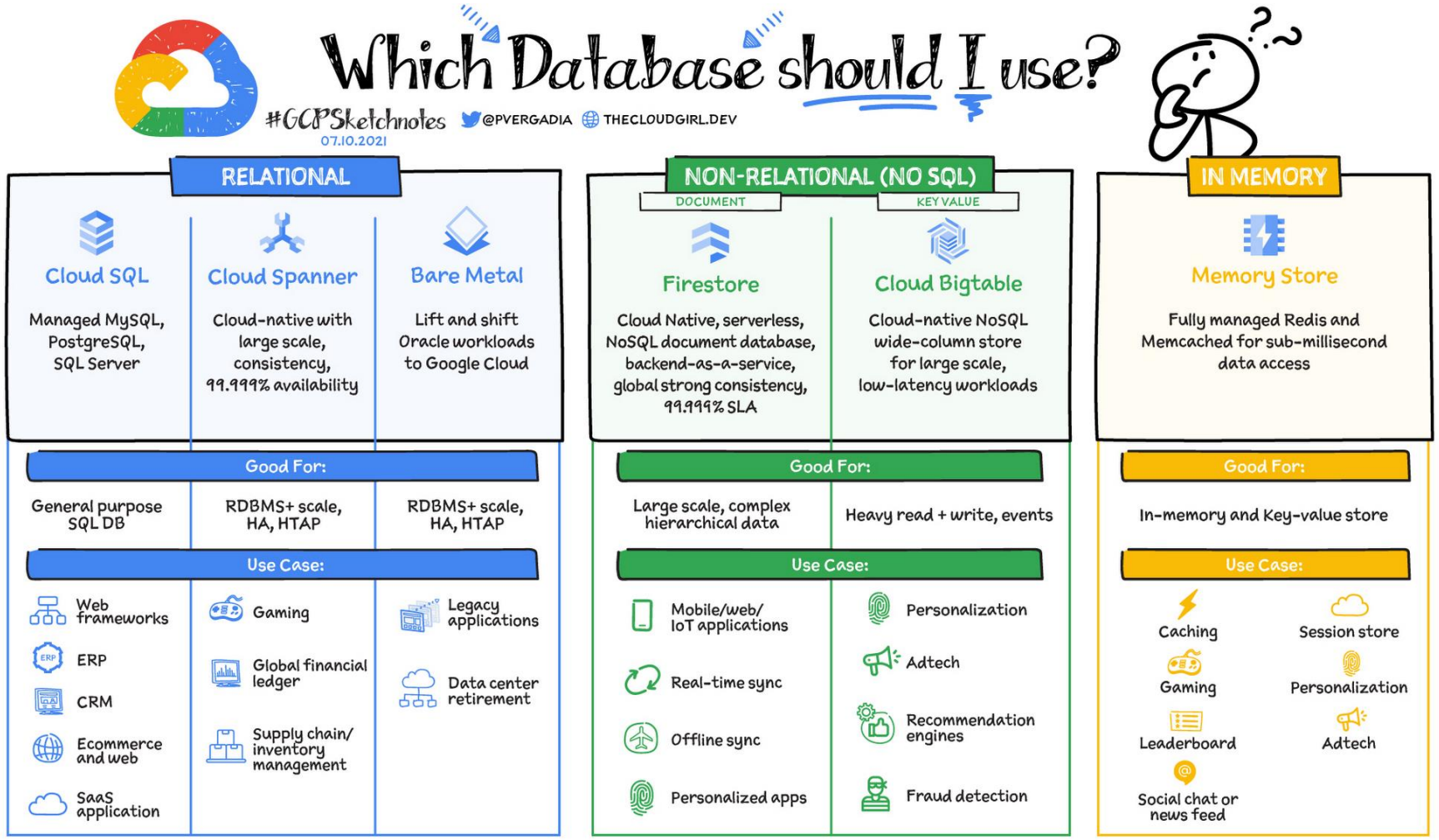
Select an Azure data store for your application



GCP data decision tree

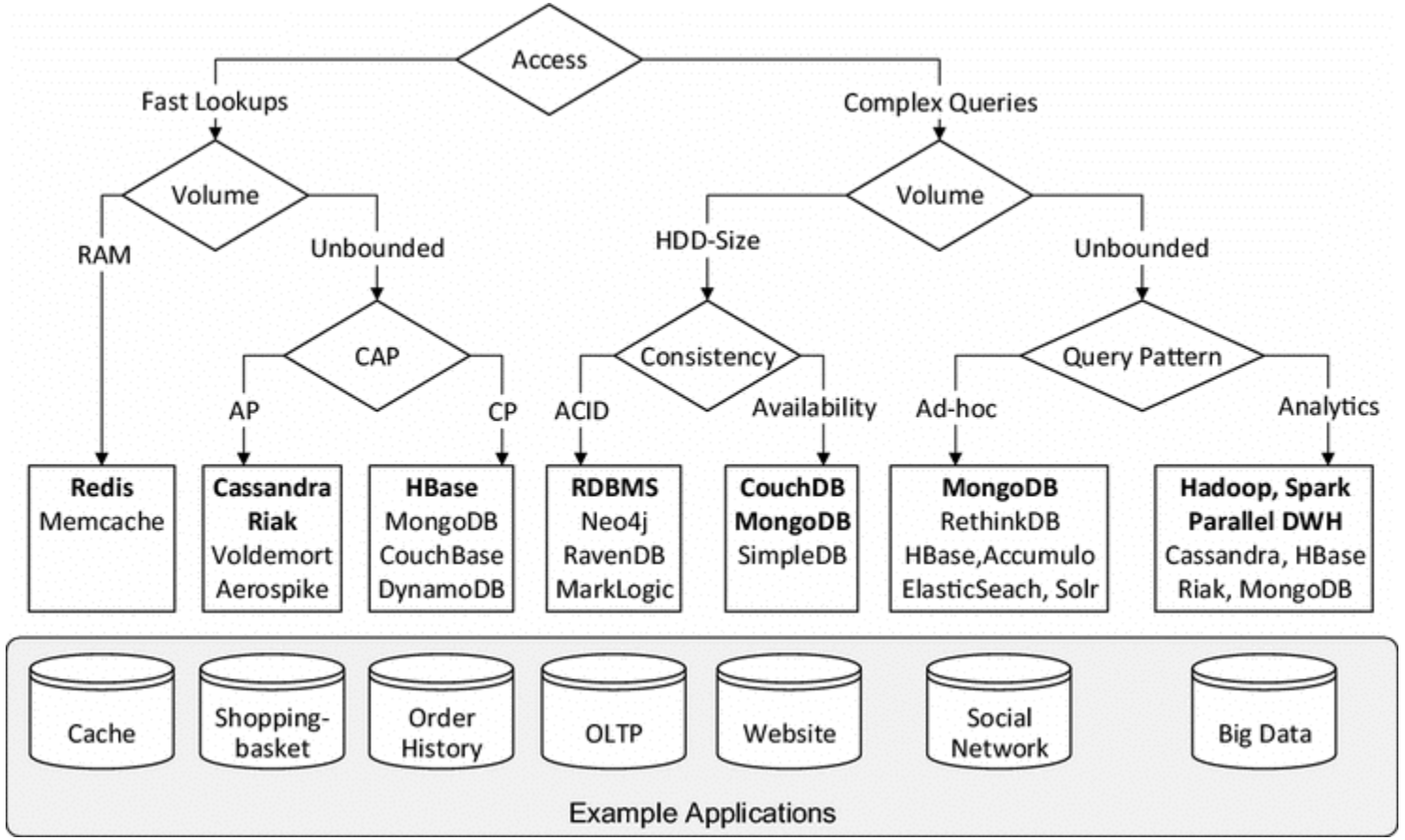


GCP data decision tree



The Choice is Yours

- Storage Type
- CAP Needs
- Popularity
- Cost
- Developer's Preference



Datastore Options

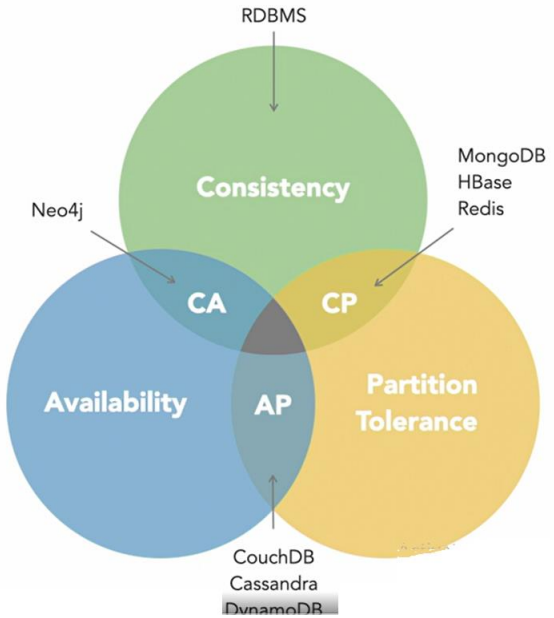
CAP Consideration

•**Consistency** means that any read request will return the most recent write. Data consistency is usually “strong” for SQL databases and for NoSQL database consistency may be anything from “eventual” to “strong”.

•**Availability** means that a non-responding node must respond in a reasonable amount of time. Not every application needs to run 24/7 with 99.999% availability but most likely you will prefer a database with higher availability.

•**Partition tolerance** means the system will continue to operate despite network or node failures.

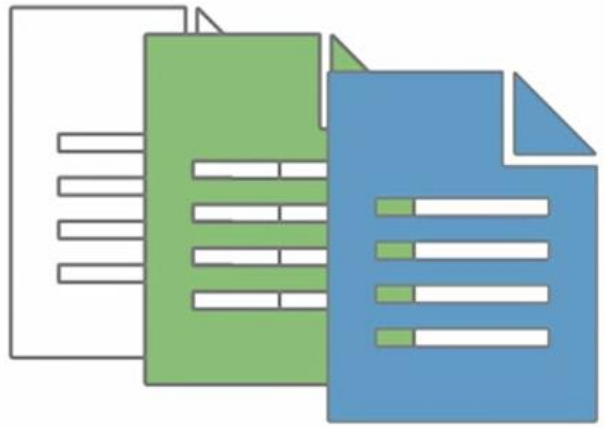
CAP Pick	NoSQL Data Store	Potential Databases
AP	Key-Value	DynamoDB
AP	Document	CouchDB
AP	Wide-Column	Cassandra
CA	Graph	Neo4J
CP	Key-Value	Redis
CP	Document	MongoDB
CP	Wide-Column	HBASE



Traditional relational databases are a natural fit for the **CA** side whereas **Non-relational database** engines mostly satisfy **AP and CP** requirements.

Document Assessment

- Semistructured
- Use hierarchical JSON-like structures
- Related data stored together rather than joined
- For example, MongoDB



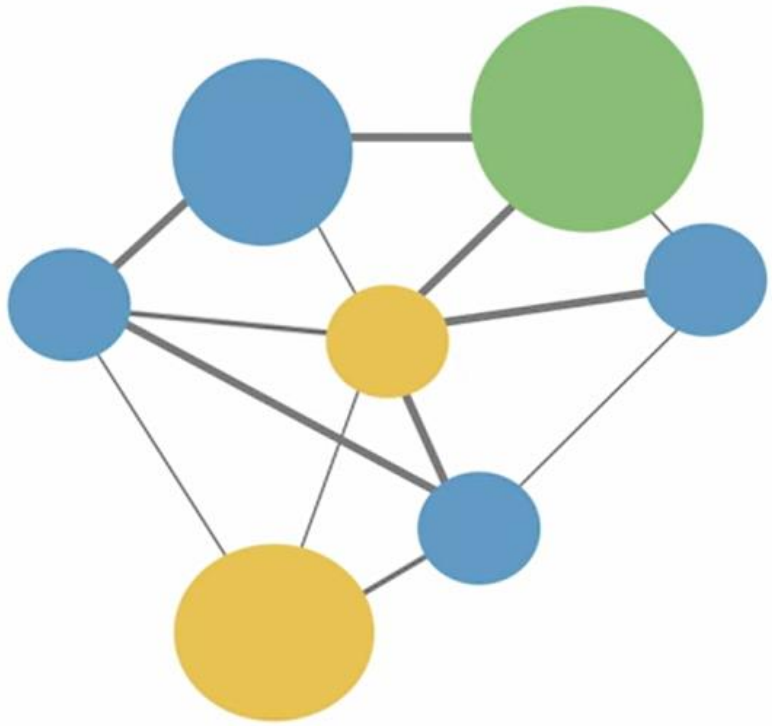
```
customer: “dan sullivan”
address: “123 front street”
```

Document Assessment - MongoDB

Strengths	Shortcomings	Applications
• Complex data types • Rich querying • Full text search • Scalability	• No transactions • No media storage • Joins not optimal • No attribute-based security	• Blogs • Catalogs • RDBMS alternative • Searchable repository

Graph Assessment

- Structured to semistructured
- Modeled on directed graph
- Nodes and edges
- Properties on nodes and edges
- For example, Neo4j



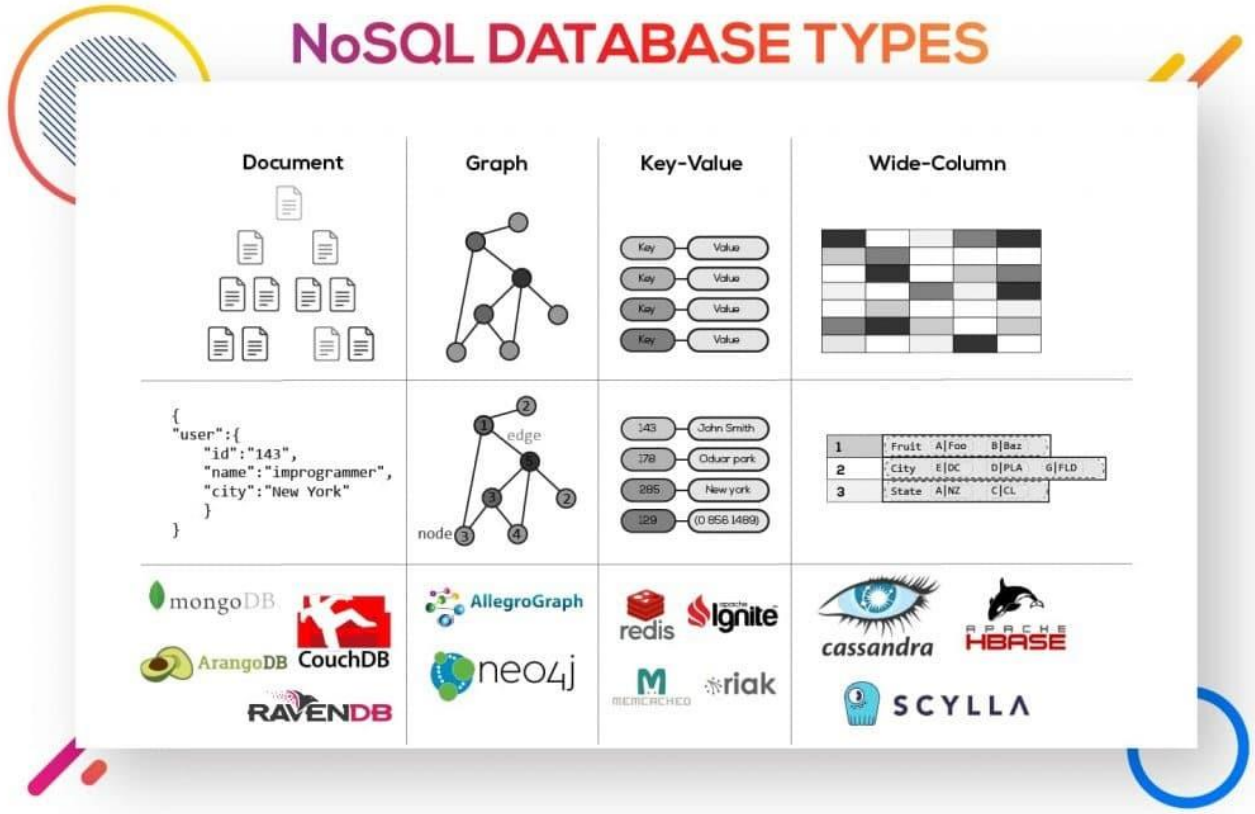
Graph Assessment – Neo4j

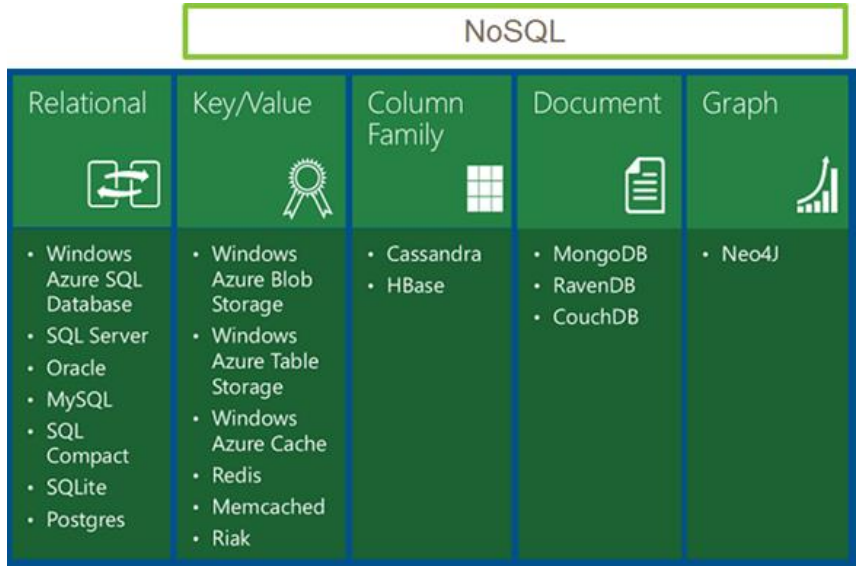
Strengths	Shortcomings	Applications
• Transactions • Relation-based queries • Deep/complex relationships • Relation attributes	• Complex data • No aggregations • No subqueries • Large volume queries	• Social media • Network topology • Recommendation engines • Location services • Search

HDFS Assessment - Hadoop

Strengths	Shortcomings	Applications
• Linear scaling • Redundancy • Security • High availability	• No updates • Limited querying • Queries – very slow • Not for small data	• Raw dumps • Media storage • Data backups

NOSQL Types


NoSQL DATABASE TYPES
Document
Graph
Key-Value
Wide-Column
{
"user":{
"id":"143",
"name":"improgrammer",
"city":"New York"
}
}
edge
node
mongoDB
CouchDB
ArangoDB
RAVENDB
AllegroGraph
neo4j
redis
Ignite
riak
cassandra
HBASE
SCYLLA


NoSQL
Relational
Key/Value
Column Family
Document
Graph
Windows Azure SQL Database
SQL Server
Oracle
MySQL
SQL Compact
SQLite
Postgres
Windows Azure Blob Storage
Windows Azure Table Storage
Windows Azure Cache
Redis
Memcached
Riak
Cassandra
HBase
MongoDB
RavenDB
CouchDB
Neo4J

Multi-model databases

- AllegroGraph – document (JSON, JSON-LD), graph
- ArangoDB – document (JSON), graph, key–value
- ArcadeDB – document (JSON), graph, key–value, time-series, SQL, Cypher query language, Gremlin (query language)
- Cosmos DB – document (JSON), graph,[6] key–value, SQL
- Couchbase – document (JSON), key–value, N1QL
- Datastax – key–value, tabular, graph
- EnterpriseDB – document (XML and JSON), key–value
- MarkLogic – document (XML and JSON), graph triplestore, binary, SQL
- Microsoft Azure SQL Database - relational, document (JSON), graph, XML
- Oracle Database – relational, document (JSON and XML), graph triplestore, property graph, key–value, objects
- OrientDB – document (JSON), graph, key–value, reactive, SQL
- PostgreSQL – relational, document (JSON and XML), key–value, graph, arrays, objects
- Redis – key–value, document (JSON), property graph, streaming, time-series
- SAP HANA – relational, document (JSON), graph, streaming
- Virtuoso Universal Server – relational, document (XML), RDF graphs

Comparison of multi-model databases

Database	SQL	Document	Graph	Object	License	Transactions
ArcadeDB	Yes	Yes	Yes	Yes	Apache 2 License	Full ACID
ArangoDB	No	Yes	Yes	No	Apache 2 License	Full ACID, pessimistic locking, configurable durability
Azure Cosmos DB	Yes	Yes	Yes	Yes	Proprietary	Full ACID within a partition, multiple consistency models
CrateDB	Yes	Yes	No	Yes	Apache 2 License	Eventual consistency, Optimistic concurrency control
EnterpriseDB	Yes	Yes	No	Yes	Proprietary	Full ACID
MarkLogic	Yes	Yes	Yes	No	Proprietary	Full ACID
OrientDB	Yes	Yes	Yes	Yes	Apache 2 License	Full ACID, even distributed
SAP HANA	Yes	Yes	Yes	No	Proprietary	Full ACID
SAP HANA Cloud	Yes	Yes	Yes	No	Proprietary	Full ACID
Virtuoso	Yes	Yes	Yes	Yes	Proprietary *or* GNU GPL v2	Full ACID

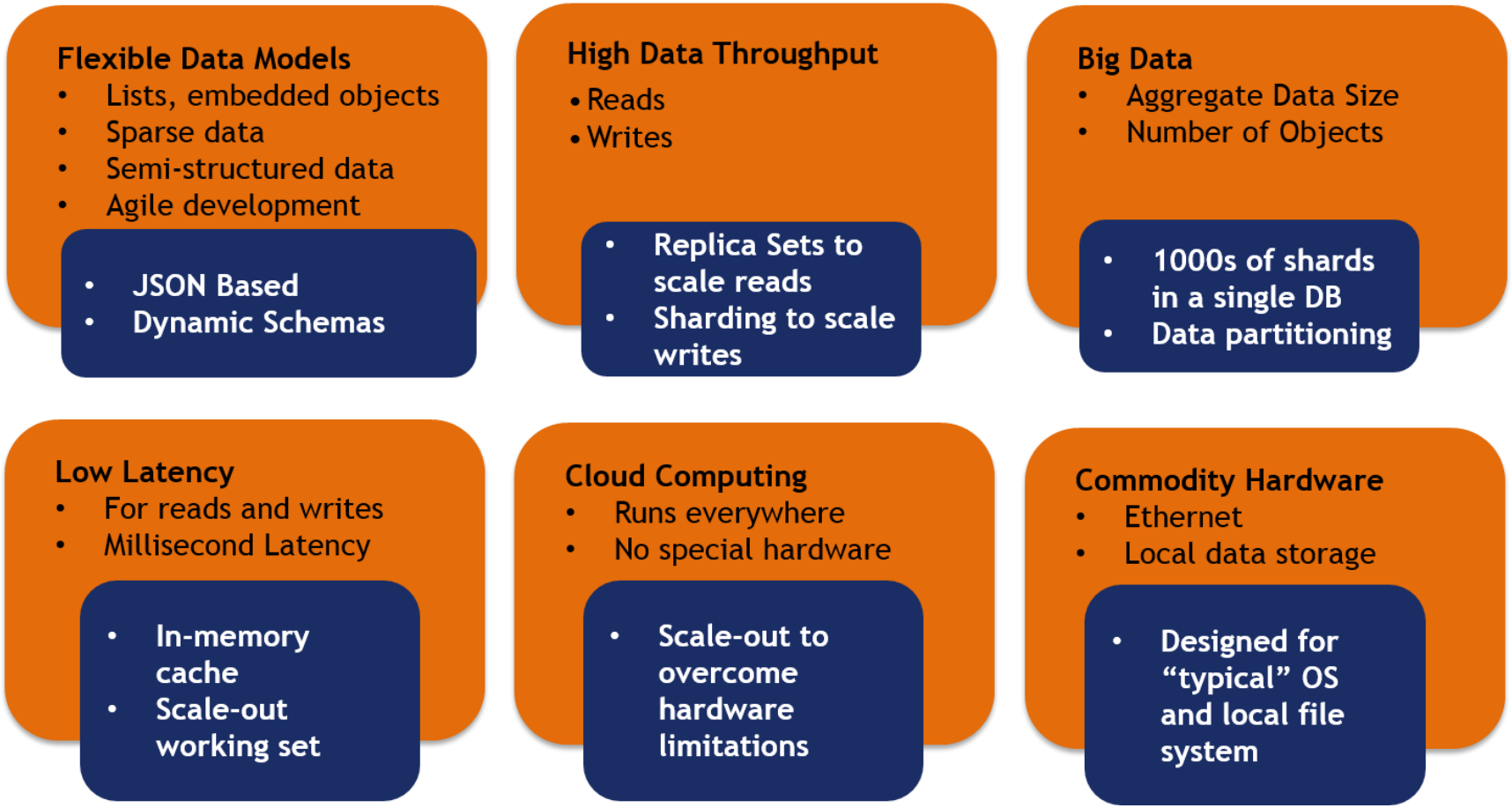
JSON - BSON

```
{
  "_id": 1,
  "name": { "first" : "John", "last" : "Backus" },
  "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards": [
    {
      "award": "W.W. McDowell Award",
      "year": 1967,
      "by": "IEEE Computer Society"
    }, {
      "award": "Draper Prize",
      "year": 1993,
      "by": "National Academy of Engineering"
    }
  ]
}
```

{"hello": "world"} →
\x16\x00\x00\x00 // total document size
\x02 // 0x02 = type String
hello\x00 // field name
\x06\x00\x00\x00world\x00 // field value
\x00 // 0x00 = type EOO ('end of object')

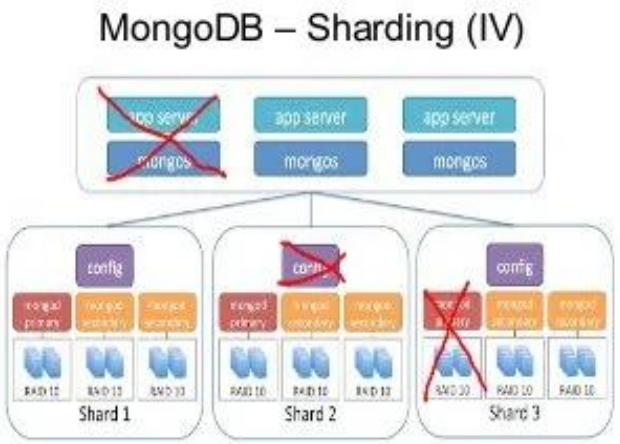
{"BSON": ["awesome", 5.05, 1986]} →
\x31\x00\x00\x00
\x04BSON\x00
\x26\x00\x00\x00
\x02\x30\x00\x08\x00\x00\x00awesome\x00
\x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40
\x10\x32\x00\xc2\x07\x00\x00
\x00
\x00

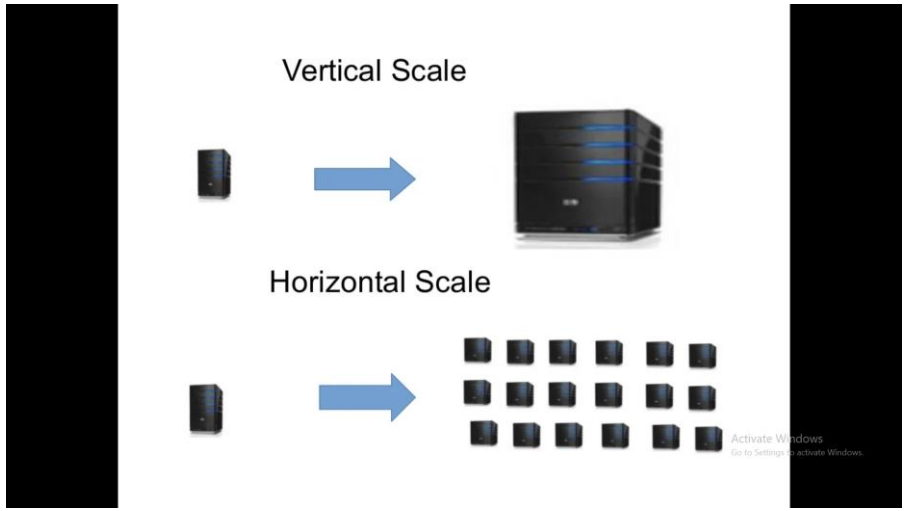
NOSQL - Advantages



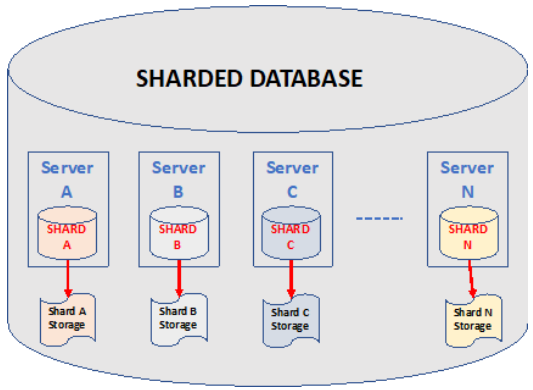
NOSQL – Some Features

Auto-Sharding
Replication
Caching
Dynamic
Schema





NOSQL – Some Features Sharding



Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	CITY
1	Alice	Anderson	Austin
2	Bob	Best	Boston
3	Carrie	Conway	Chicago
4	David	Doe	Denver

Vertical Shards

VS1

CUSTOMER ID	FIRST NAME	LAST NAME
1	Alice	Anderson
2	Bob	Best
3	Carrie	Conway
4	David	Doe

VS2

CUSTOMER ID	CITY
1	Austin
2	Boston
3	Chicago
4	Denver

Horizontal Shards

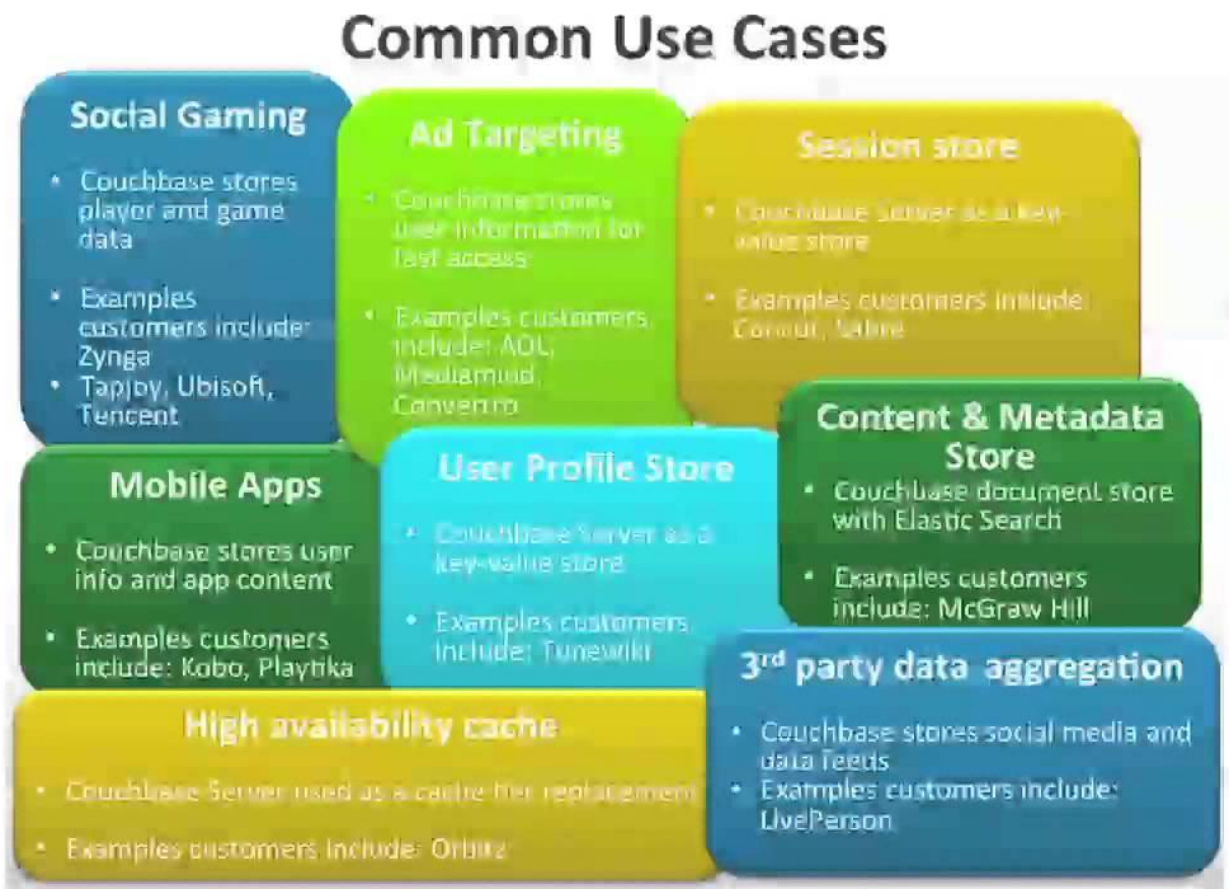
HS1

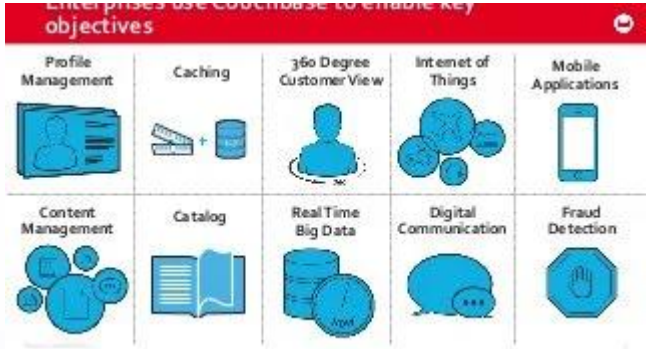
CUSTOMER ID	FIRST NAME	LAST NAME	CITY
1	Alice	Anderson	Austin
2	Bob	Best	Boston

HS2

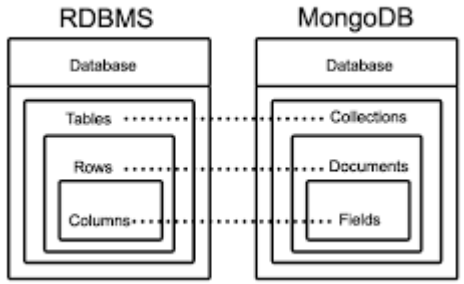
CUSTOMER ID	FIRST NAME	LAST NAME	CITY
3	Carrie	Conway	Chicago
4	David	Doe	Denver

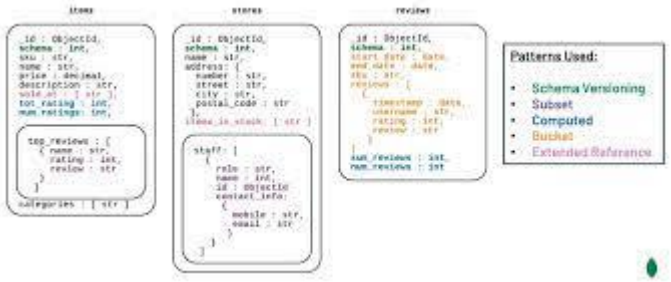
NOSQL - Common Use Case





MongoDB - Modelling





One-to-One	User works for a single company (embedding)	*User*
		{
		"_id": "ObjectId('AAA')",
		"name": "Joe Karlsson",
		"company": "MongoDB"
		}

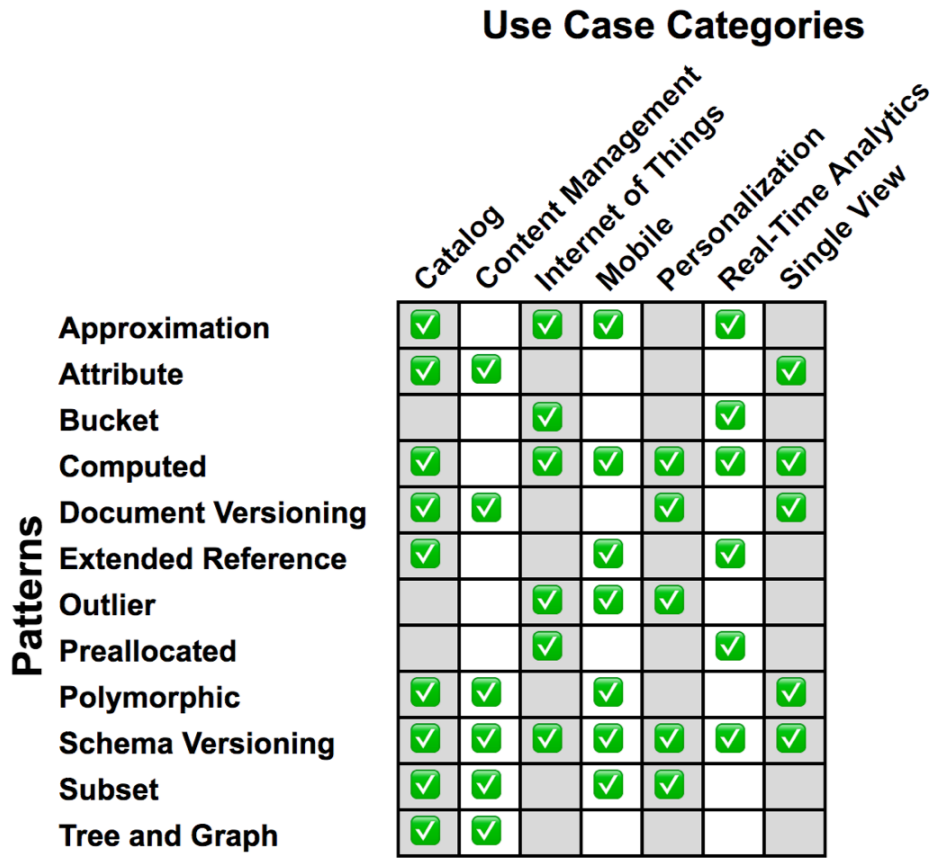
One-to-Few	User has a few addresses (embedding)	*User*
		{
		"_id": "ObjectId('AAA')",
		"name": "Joe Karlsson",
		"company": "MongoDB",
		"addresses": [
		{ "street": "45 Sesame St", "city": "Los Angeles" },
		{ "street": "123 Avenue Q", "city": "New York" }
		]
		}

One-to-Many	Product has many parts (child referencing)	*Product*
		{
		"name": "left-handed smoke shifter",
		"manufacturer": "Acme Corp",
		"catalog_number": "1234",
		"parts": [
		"ObjectID('AAAA')",
		"ObjectID('BBBB')",
		"ObjectID('CCCC')"
		]
		}
		
		Part
		{
		"_id" : "ObjectID('AAAA')",
		"partno" : "123-aff-456",
		"name" : "#4 grommet",
		"qty": "94",
		"cost": "0.94",
		"price":" 3.99"
		}

One-to-Huge	Host server has millions of log messages (parent referencing)	*Host*
		{
		"_id": ObjectID("AAAB"),
		"name": "goofy.example.com",
		"ipaddr": "127.66.66.66"
		}
		
		Log
		{
		"time": ISODate("2014-03-28T09:42:41.382Z"),
		"message": "cpu is on fire!",
		"host": ObjectID("AAAB")
		}

Many-to-Many	User has many tasks and a task can be assigned to many users (mutual referencing)	*User*
		{
		"_id": ObjectID("AAF1"),
		"name": "Kate Monster",
		"tasks": [
		ObjectID("ADF9"),
		ObjectID("AE02"),
		ObjectID("AE73")
		]
		}
		
		Task
		{
		"_id": ObjectID("ADF9"),
		"description": "Write blog post about schema design",
		"due_date": ISODate("2014-04-01"),
		"owners": [ObjectID("AAF1"), ObjectID("BB3G")]
		}

MongoDB –Patterns & Use Case Categories



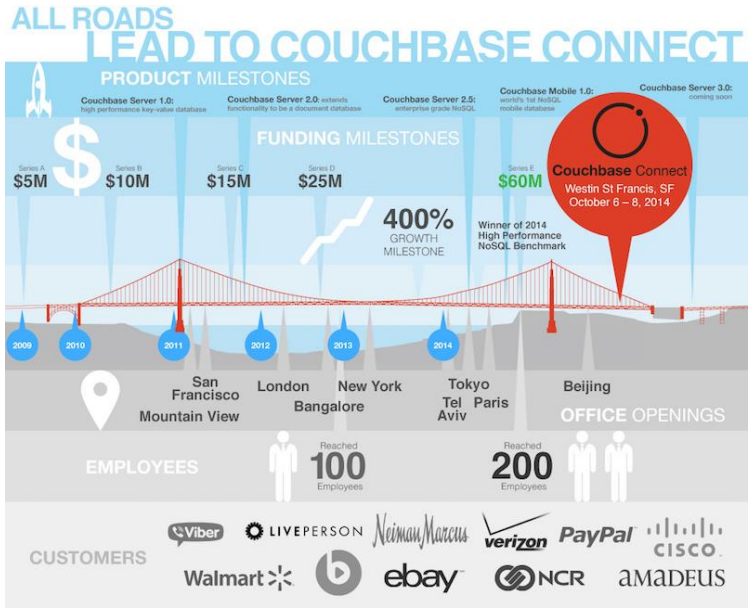
Real-life Examples Of MongoDB Use Cases

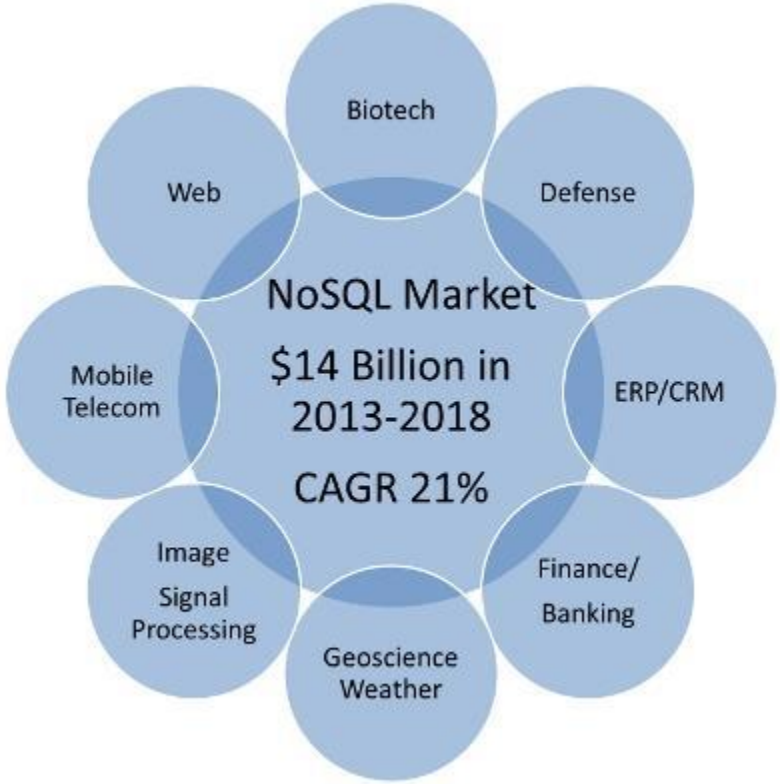
- **1) India's Aadhaar**
- Aadhaar is a fantastic example of a real-world MongoDB use case. Aadhaar is India's Unique Identification Project and the world's most extensive Biometrics Database System. The program launched in 2009 has collected demographic and biometric information from over 1.2 billion people. Aadhaar relied on MongoDB, among other database systems like HBase, MySQL, and Hadoop, to store this massive amount of data. However, MongoDB was one of the database systems first purchased to power the search strategy.
- **2) Forbes**
- Another real-world MongoDB use case is Forbes. When a story becomes viral, people resort to every available website to get details. Therefore, it is essential for publishers to be alert and give shareable information as soon as possible to keep their readers informed.
- This prompted Forbes to find practical solutions to boost its Content Engagement Rate. Forbes constructed its CMS and mobile application in two months using MongoDB. They had to redesign their website and switch to MongoDB to upload content from anywhere in the world efficiently.
- Their editors also took advantage of MongoDB's flexible structure to deliver dynamic quality material promptly to their viewers. This brave action eventually paid off. With this, they could eliminate poor existing practices while lowering overhead expenses.
- **3) MetLife**
- MetLife is a market leader in employee insurance, benefit plans, and pensions. They serve over 90 million clients across the Middle East, Europe, Asia, Latin America, Japan, and the United States. MetLife's sophisticated customer service solution, "**The Wall," is built on MongoDB**.
- The Wall is a tool that offers a consolidated view of MetLife Customers' payments, policy statements, and other information. It functions like Facebook's Wall, which gathers data from 70 historical systems and integrates it into a database table. The Wall has a capacity of 24 TB and is distributed across six different servers in two Data Management Centers. Currently, MetLife is looking to further improve its service with several Big Data initiatives, including MongoDB-based apps.
- **4) Otto**
- Many retail businesses need Real-time Analytics. This requires that their apps remain relevant and always available online, with no interruption. This is because slow response time leads to a lot of revenue loss in the retail industry due to the cutthroat competition.
- When consumers have several alternatives, they will not hesitate to switch to a more reliable service provider. For example, Otto is a major e-commerce company with a massive user base looking to meet and exceed its customers' expectations. However, prompt response time was a significant challenge because their website has over 500 companies.
- Otto is a classic MongoDB use case where they used MongoDB to **lower their response time to between 1 to 2 seconds** to solve this problem. They had to rebuild their entire catalog application. They used MongoDB because of its customizable schema capabilities, reliability, and growth characteristics.

Real-life Examples Of MongoDB Use Cases

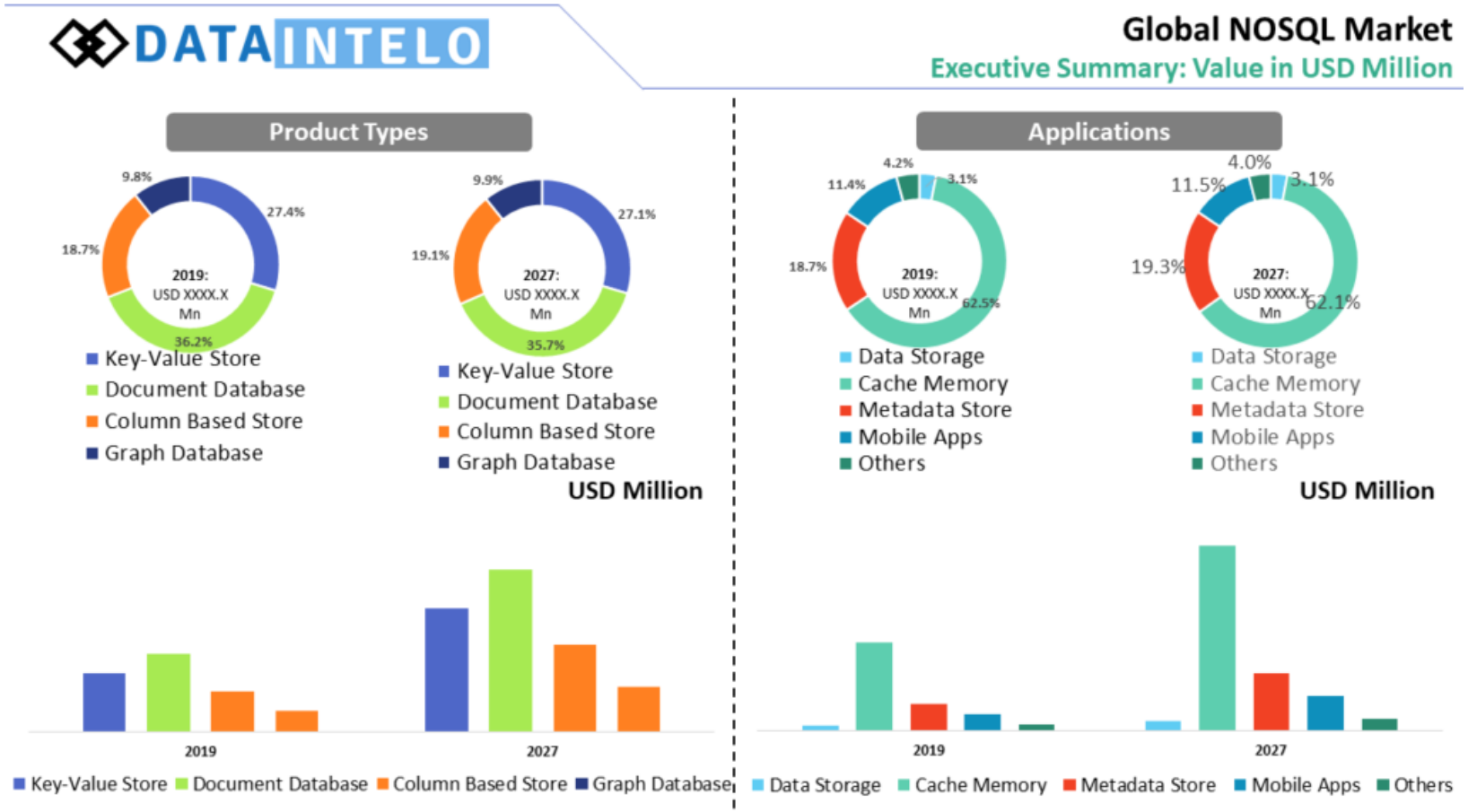
- **5) Shutterfly**
- Shutterfly is a major **Digital Picture Exchange and Private Publishing Firm** with over 6 billion photographs and a processing volume of up to 10,000 calculations every second. It is one of the businesses that switched from Oracle to MongoDB.
- During their transition to MongoDB, it became clear that running non-Relational Databases would better meet the company's data demands, potentially enhancing programmer efficiency and customizability.
- Before choosing MongoDB, Shutterfly investigated several other database systems, including BerkeleyDB and Cassandra. However, the firm settled for MongoDB while implementing it to data relevant to uploaded photographs. Still, the conventional RDBMS remains in place for aspects of the process that require a fuller transactional architecture, such as payment and account administration.
- Shutterfly is pleased with its choice to adopt MongoDB, as noted by its data architect, Kenny Gorman, who said the firm believes in picking the right technology for the task. His statement showed that MongoDB was an excellent match, although not without tradeoffs. For example, many issues cropped up before they streamlined their services to match their users' needs.
- **6) FACEIT**
- Major gaming firms like FACEIT and SEGA have effectively used MongoDB to improve their consumers' interactive experiences. Underneath the shell, FACEIT relies on MongoDB as its primary Database System. In addition, MongoDB manages the orchestration of services between gamers, groups, and contests.
- FACEIT even uses MongoDB to maintain its User Profile and Tournament Data. Users' live streaming data is saved in MongoDB, and other metrics are used to measure players' activity and interaction. Due to **MongoDB's Flexible Structure and Extensive Query Model**, FACEIT could keep user profiles more effectively.
- **7) Weather Channel**
- While running its website, weather.com, the Weather Channel had problems servicing a massive clientele because it used a standard RDBMS system. However, MongoDB enabled them to develop their Smartphone application that serves over 40 million active users and provides **Real-time Meteorological Data** to its clients.
- They also employed **MongoDB's Scalable Solution and** **MapReduce Functionalities** to perform Real-time Analytics and forecasts based on weather data obtained. The prototype version that had previously taken weeks to develop was delivered in just a few hours. This gave The Weather Channel an edge over the competition.

NoSQL – Use Case

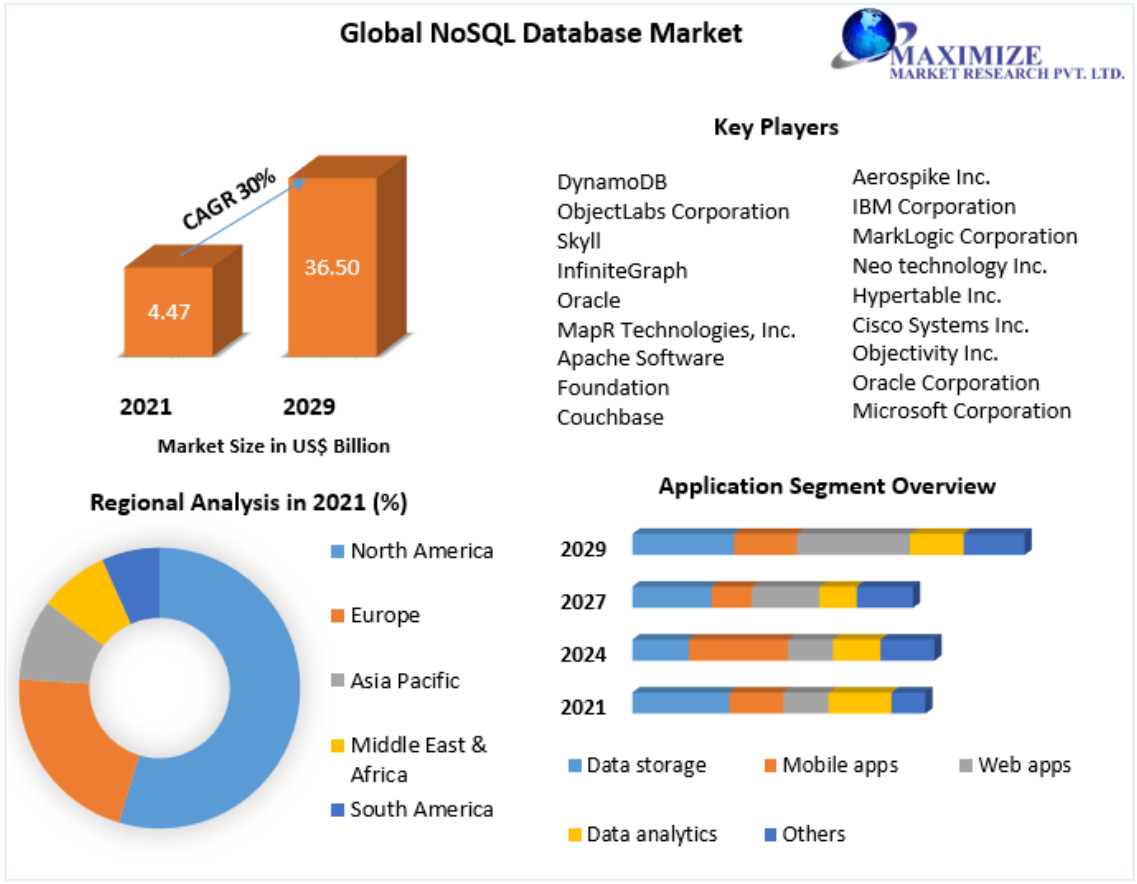




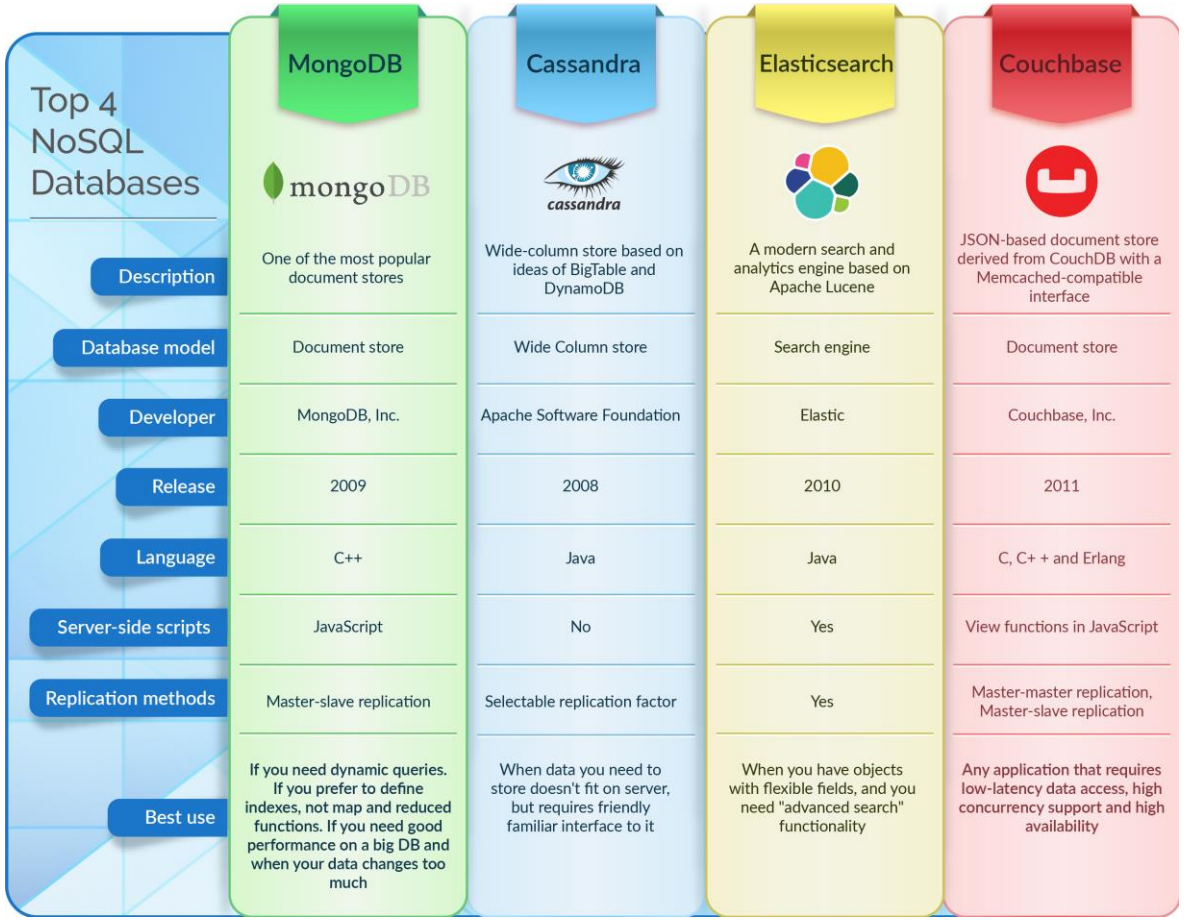
NoSQL Market



NOSQL Market



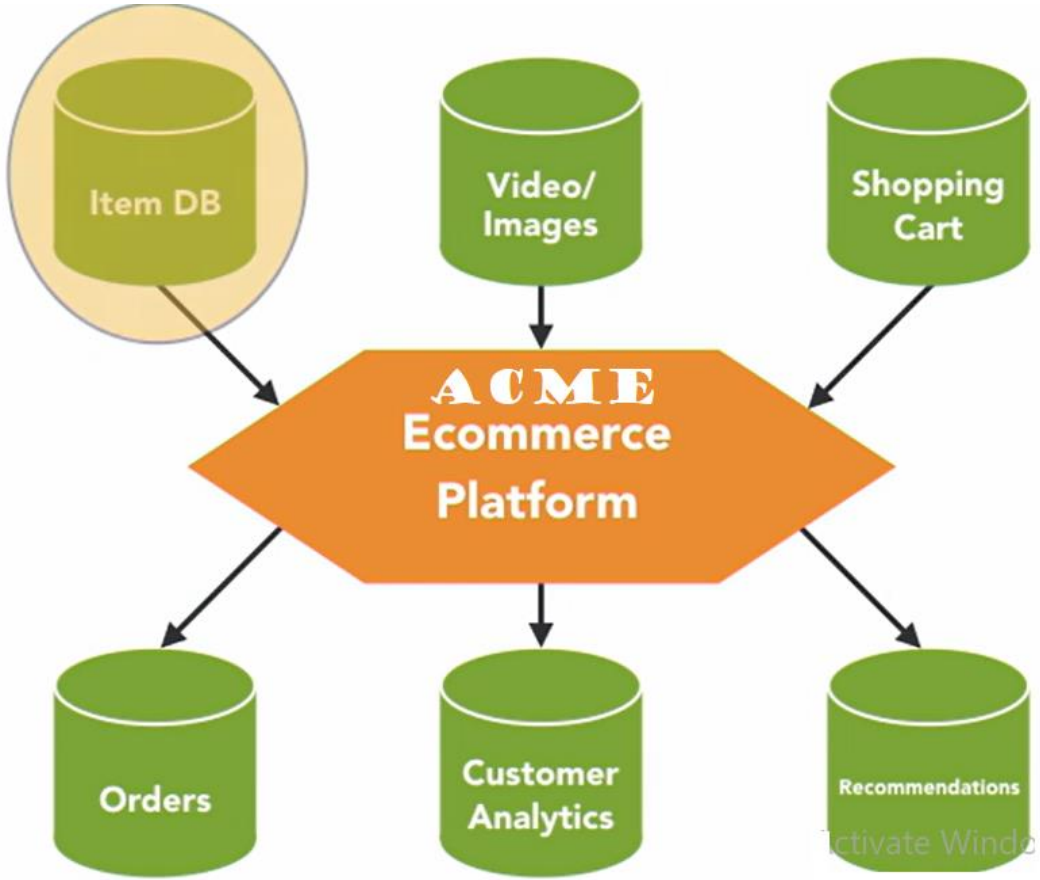
Nosql Comparision



ACME Ecommerce Platform
Utopic Database Choices

- Description, reviews, similar products
- Flexible schema
- Link to video/images

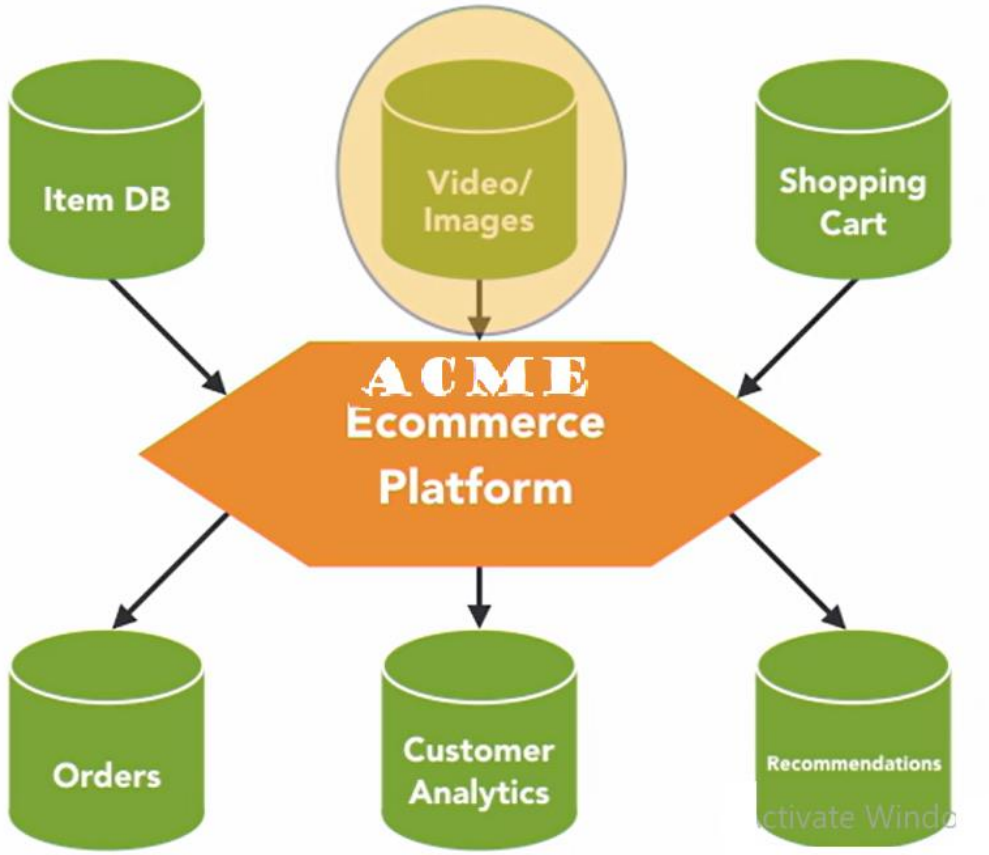
Item Database: MongoDB



ACME Ecommerce Platform
Utopic Database Choices

- Large files
- Streaming capabilities
- Link in MongoDB

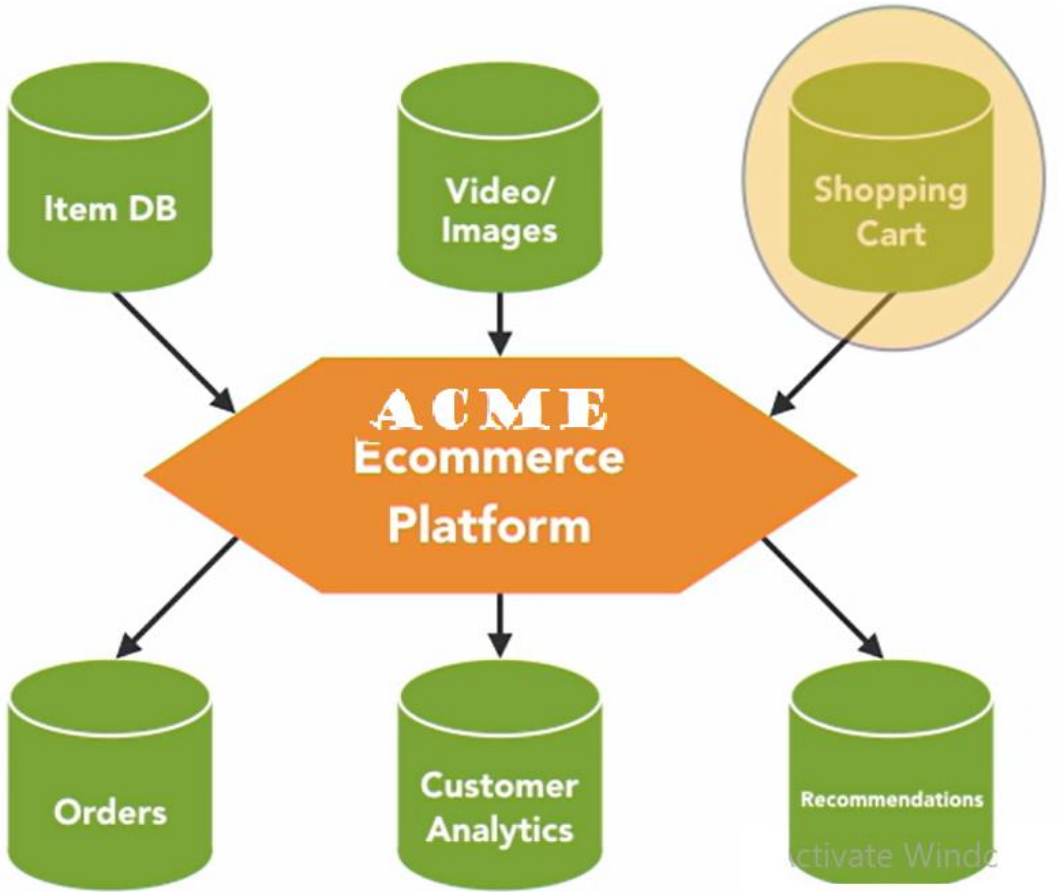
Media Database: HDFS



ACME Ecommerce Platform
Utopic Database Choices

- Shopping cart
- Key = Customer ID
- List of items
- Scalable
- Ultrafast access

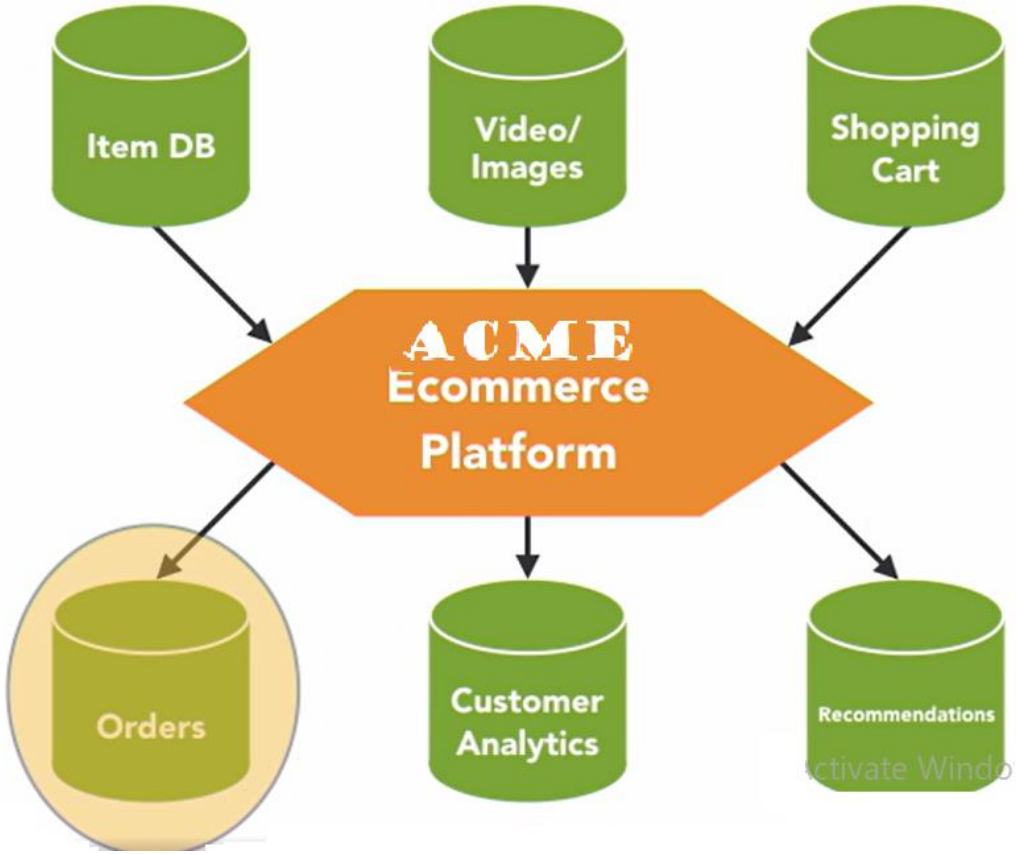
Shopping Cart: Redis



ACME Ecommerce Platform
Utopic Database Choices

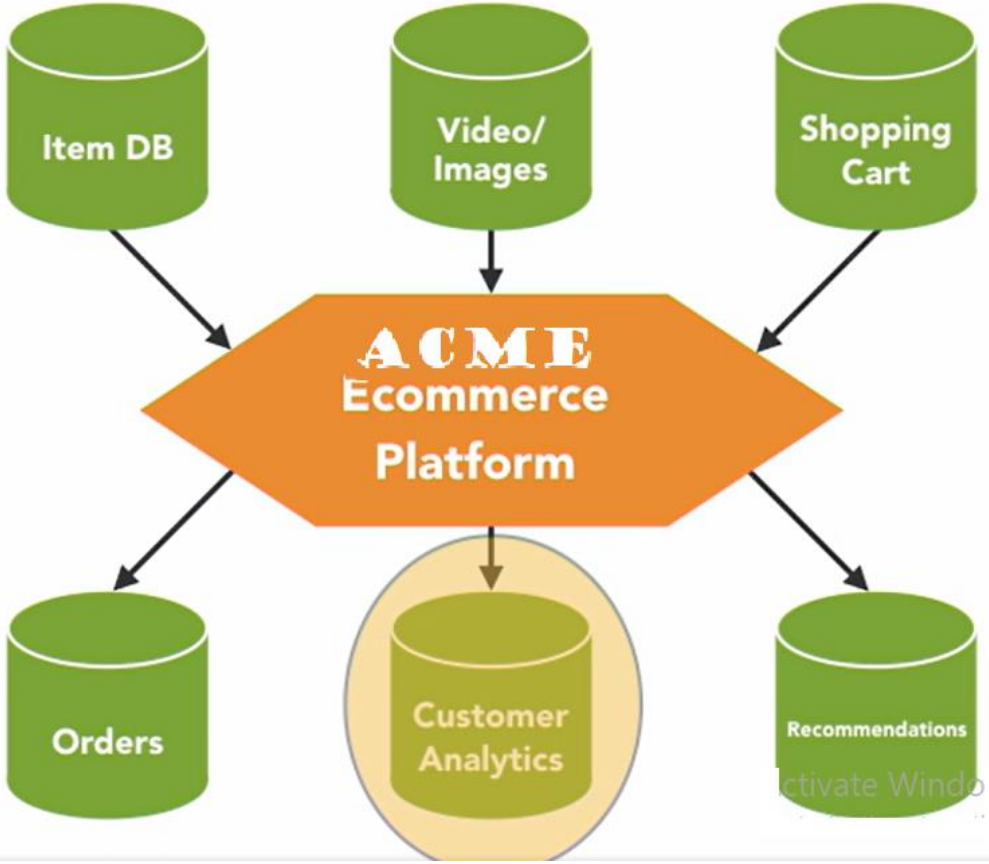
- OLTP
- Data integrity
- Transaction support
- Fast operations
- Update optimal

Orders: Oracle



ACME Ecommerce Platform
Utopic Database Choices

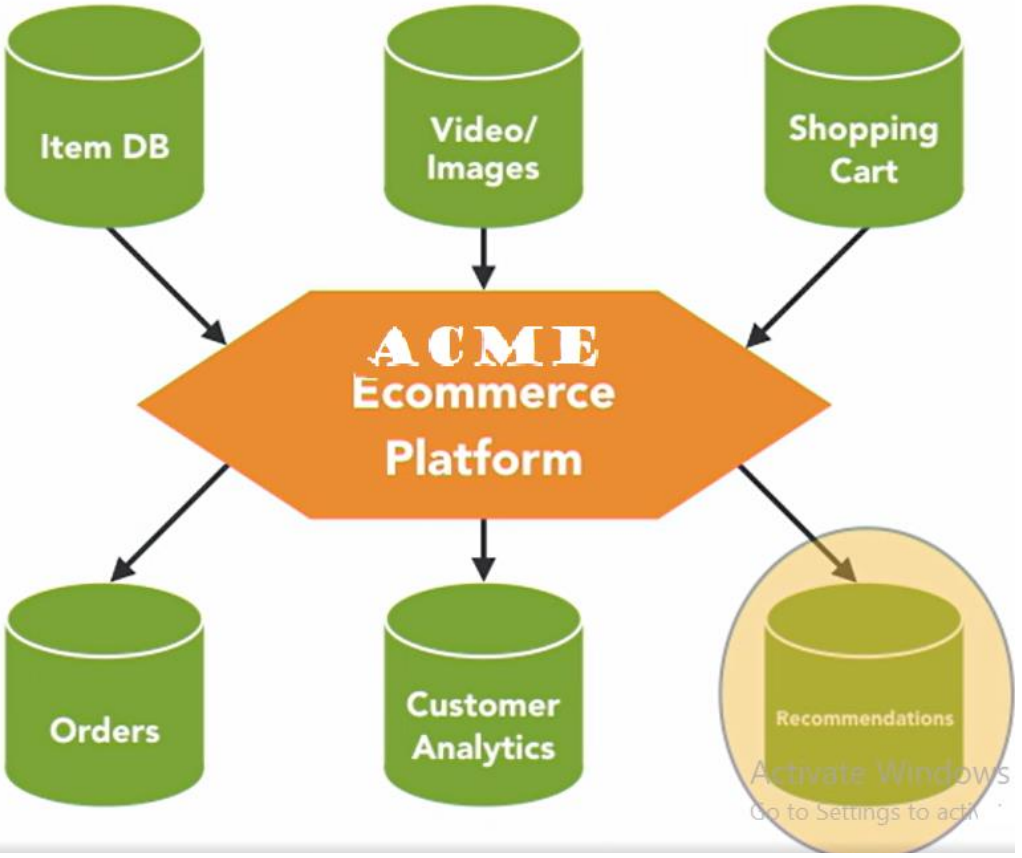
- Key = Customer
- Customer 360 view
- Summarized data
- Constant updates
- Real-time access



Customer Analytics: Cassandra

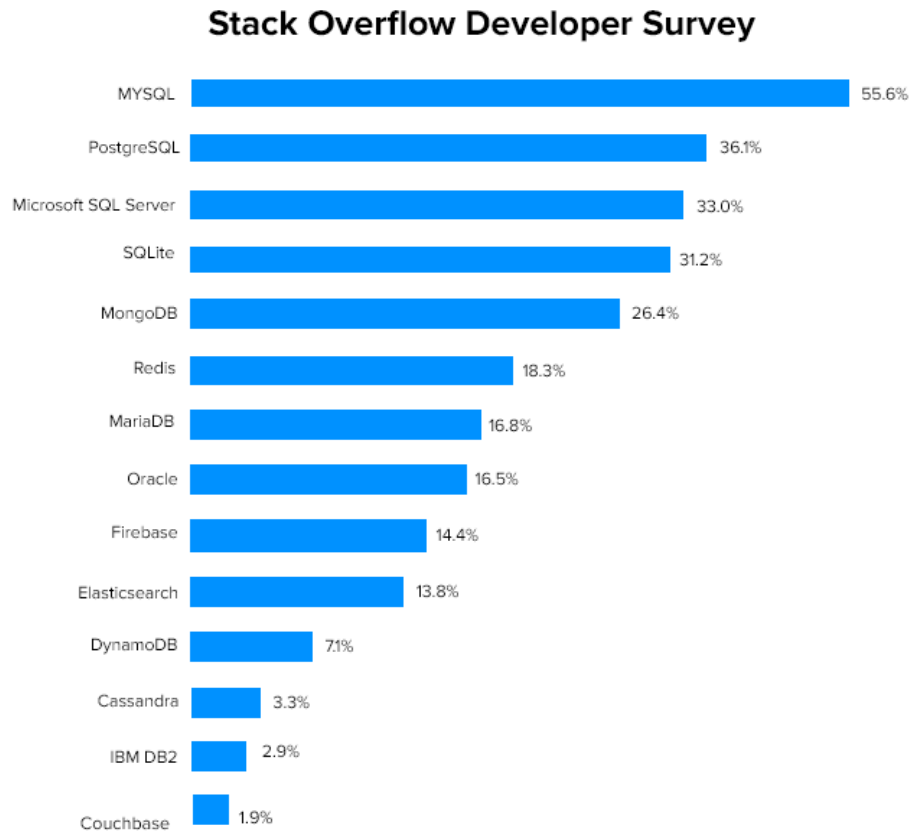
ACME Ecommerce Platform
Utopic Database Choices

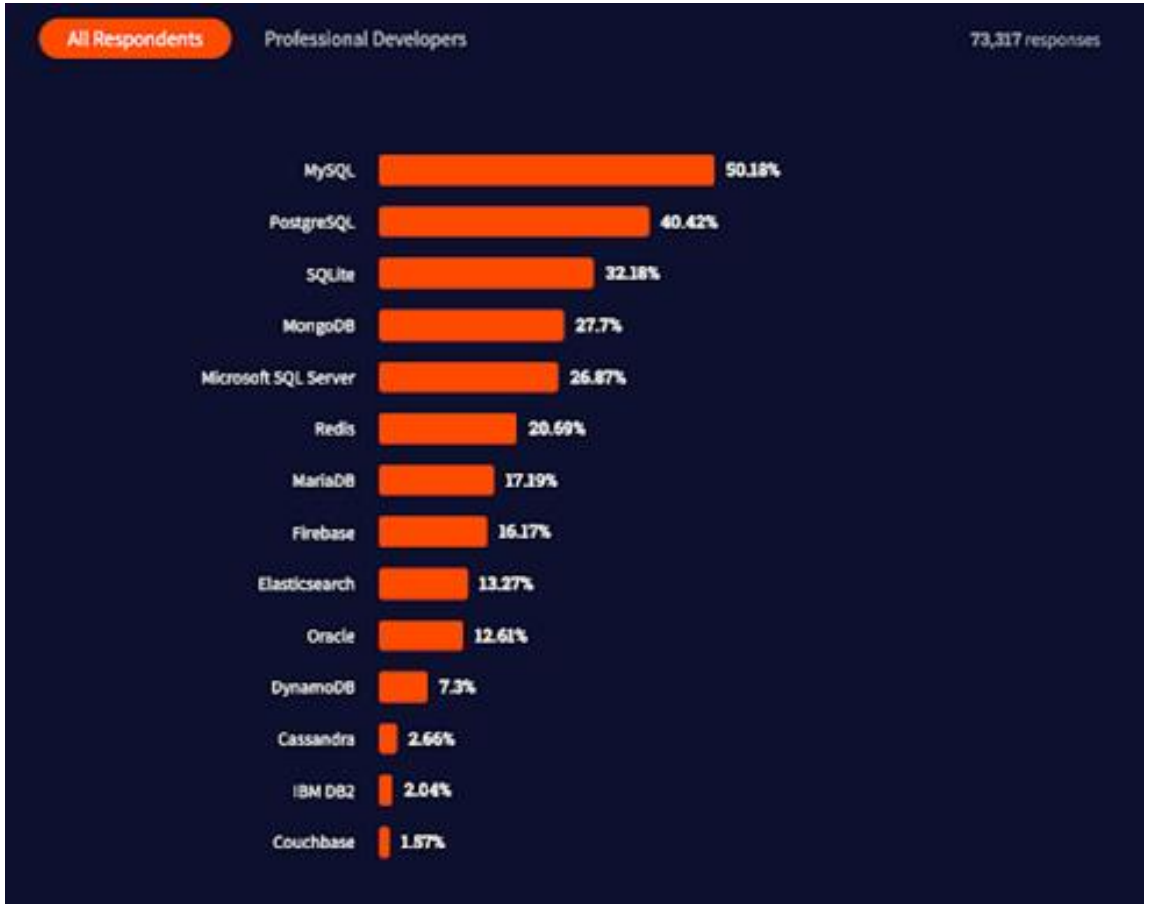
- Users and items as nodes
- Recommendations as relationships



Recommendations: Neo4j

Database Usage Metrics





Database Usage Statistics