Return to Table of Contents

Choose a Lesson

- Choosing a Managed Database
- Cloud SQL Basics
- Importing Data
- SQL Query Best Practices

Cloud SQL Basics

What is Cloud SQL?

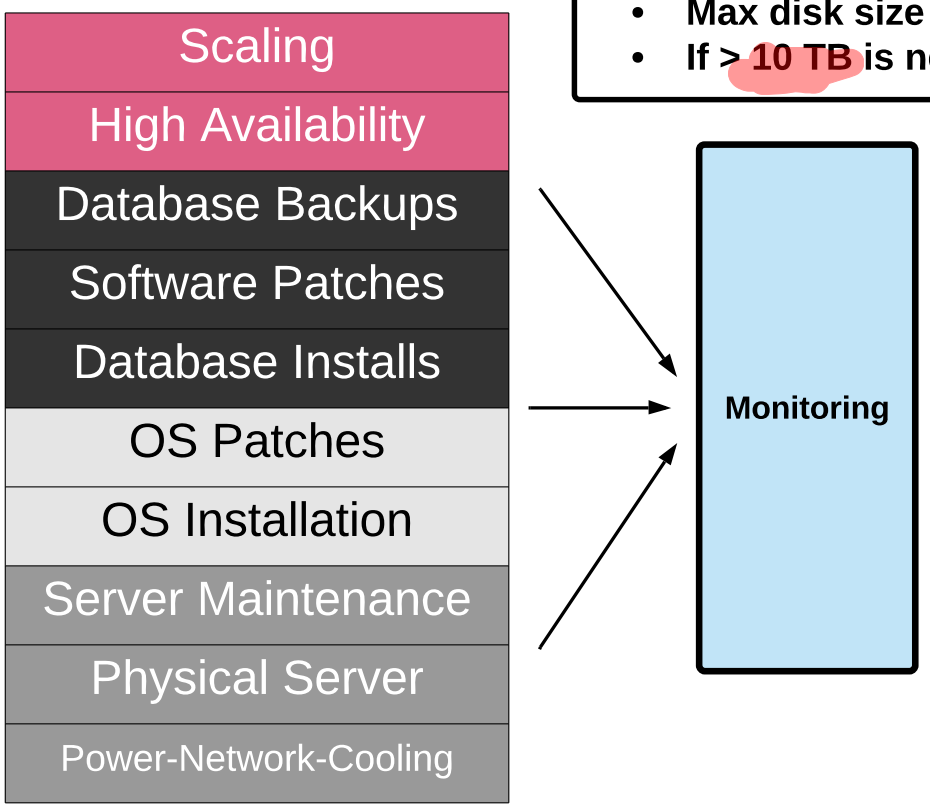
- **Direct lift and shift of traditional MySQL/PostgreSQL workloads with the maintenance stack managed for you**

What is managed?

- **OS installation/management**
- **Database installation/management**
- **Backups**
- **Scaling - disk space**
- **Availability:**
  - **Failover**
  - **Read replicas**
- **Monitoring**
- **Authorize network connections/proxy/use SSL**

Limitations:

- **Read replicas limited to the same region as the master:**
  - **Limited global availability**
- **Max disk size of 10 TB**
- **If > 10 TB is needed, or global availability in RDBMS, use Spanner**



import/export
CSV file
to instan

Return to Table of Contents

Choose a Lesson

- Choosing a Managed Database
- Cloud SQL Basics
- Importing Data
- SQL Query Best Practices

Importing Data

Importing data into Cloud SQL:

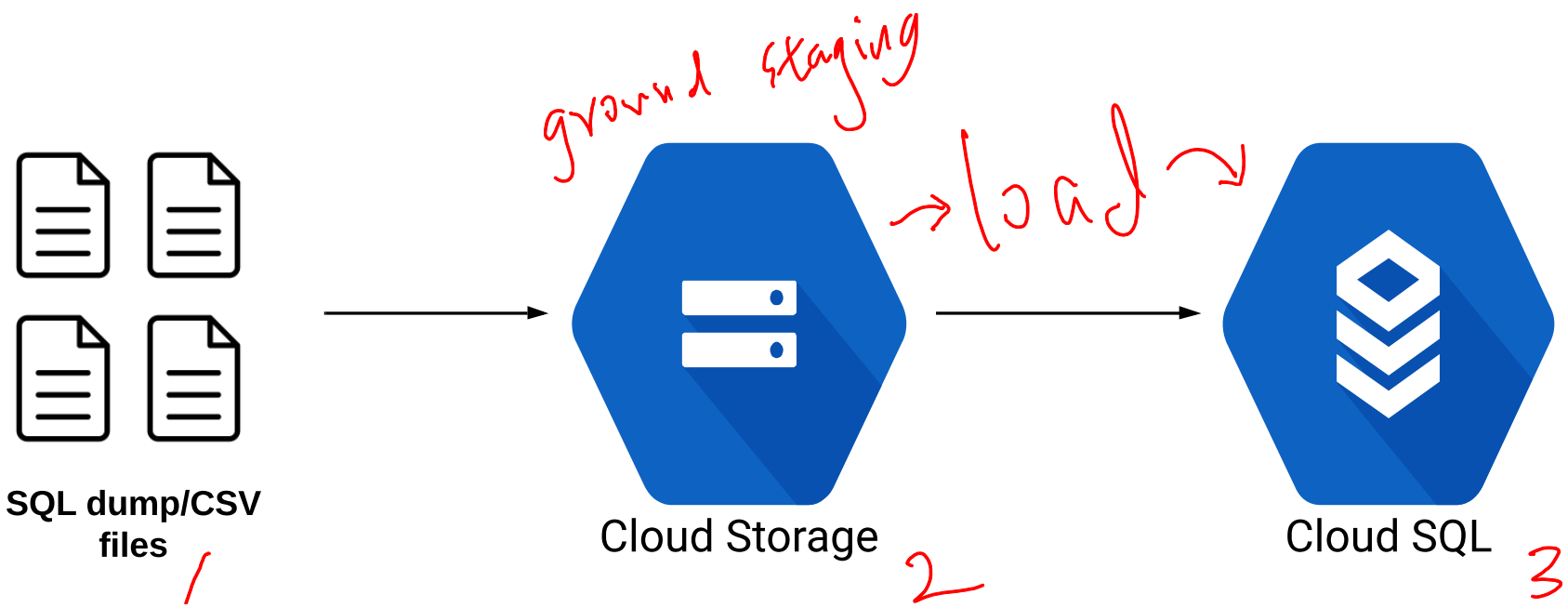
- Cloud Storage as a staging ground
- SQL dump/CSV file format

Export/Import process:

1. Export SQL dump/CSV file:
   - SQL dump file cannot contain triggers, views, stored procedures
2. Get dump/CSV file into Cloud Storage
3. Import from Cloud Storage into Cloud SQL instance

Best Practices:

- Use correct flags for dump file (--'flag_name'):
  - Databases, hex-blob, skip-triggers, set-gtid-purged=OFF, ignore-table
- Compress data to reduce costs:
  - Cloud SQL can import compressed .gz files
- Use InnoDB for Second Generation instances



Return to Table of Contents

Choose a Lesson

Choosing a Managed Database

Cloud SQL Basics

Importing Data

SQL Query Best Practices

SQL Query Best Practices

General SQL efficiency best practices:

- More, smaller tables better than fewer, large tables:
  - Normalization of tables
- Define your SELECT fields instead of using SELECT *:
  - SELECT * acts as a 'select all'
- When joining tables, use INNER JOIN instead of WHERE:
  - WHERE creates more variable combinations = more work

Questions:

SQL table grow over time

Larger table → slow query

Return to Table of Contents

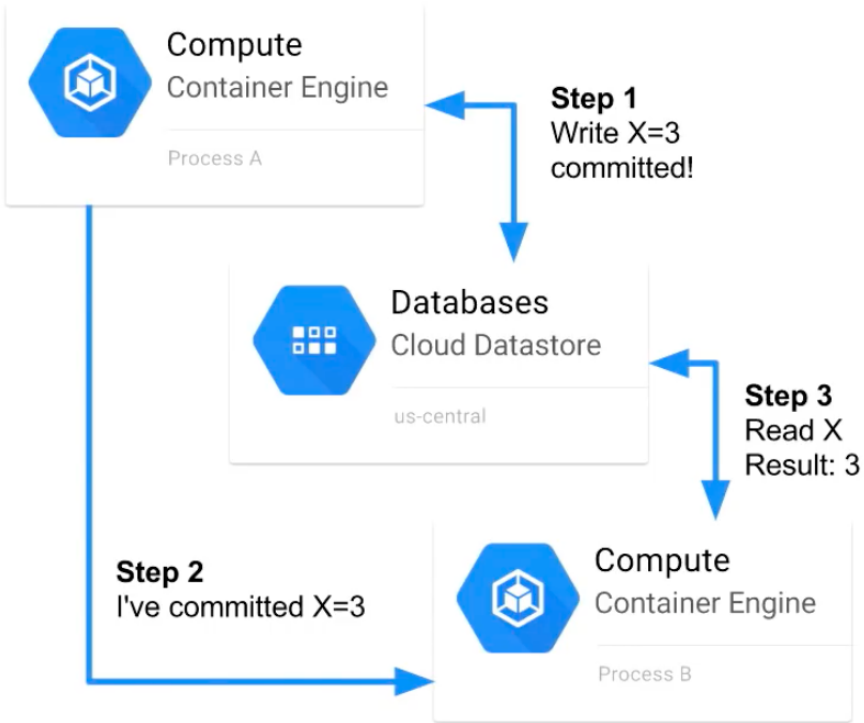
Choose a Lesson

- Cloud Datastore Overview
- Data Organization
- Queries and Indexing
- Data Consistency

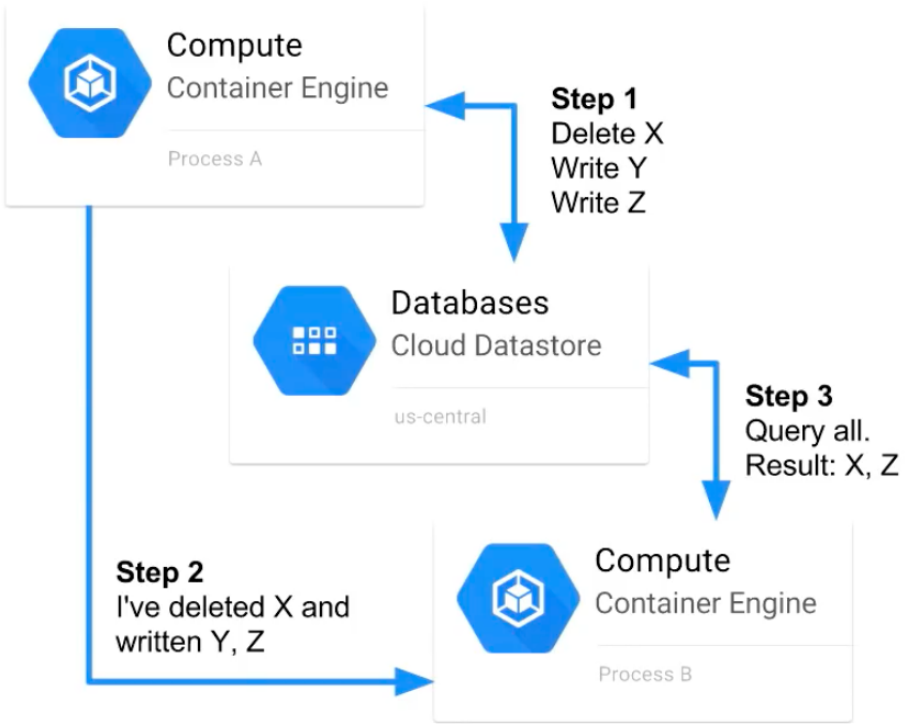
Data Consistency

Previous

Strong



Eventual



Return to Table of Contents

Choose a Lesson

- Cloud Datastore Overview
- Data Organization
- Queries and Indexing
- Data Consistency

(Cloud Datastore): Data Consistency

*Datastore is a NoOps Database

Next

What is data consistency in queries?

- "How up to date are these results?"
- "Does the order matter?"
- Strongly consistent = Parallel processes see changes in same order:
  - Query is guaranteed up to date but may take longer to complete
- Eventually consistent = Parallel process can see changes out of order, will eventually see accurate end state:
  - Faster query, but may *sometimes* return stale results
- Performance vs. accuracy
- Ancestor query/key-value operations = strong
- Global queries/projections = eventual

Use cases:

- Strong - financial transaction:
  - Make deposit -- check balance
- Eventual - census population:
  - Order not as important, as long as you get eventual result

Return to Table of Contents

Choose a Lesson

Cloud Datastore Overview

Data Organization

Queries and Indexing

Data Consistency

Queries and Indexing

Previous

Danger - Exploding Indexes!

- Default - create an entry for every possible combination of property values
- Results in higher storage and degraded performance
- Solutions:
  - Use a custom index.yaml file to narrow index scope
  - Do not index properties that don't need indexing

Return to Table of Contents

Choose a Lesson

- Cloud Datastore Overview
- Data Organization
- Queries and Indexing
- Data Consistency

Queries and Indexing

Next

Query:

- Retrieve an entity from Datastore that meets a set of conditions
- Query includes:
  - Entity kind (table)
  - Filters ← "where" in SQL
  - Sort order
- Query methods:
  1. Programmatic
  2. Web console
  3. Google Query Language (GQL)

Indexing:

- Queries gets results from indexes:
  - Contain entity keys specified by index properties
  - Updated to reflect changes
  - Correct query results available with no additional computation needed

Index types:

- Built-in - default option:
  - Allows single property queries
- Composite - specified with an index configuration file (index.yaml):
  - gcloud datastore create-indexes index.yaml

```
indexes:
- kind: Task
  properties:
  - name: tags
  - name: created
- kind: Task
  properties:
  - name: collaborators
  - name: created
```

example

Return to Table of Contents

Choose a Lesson

Cloud Datastore Overview

Data Organization

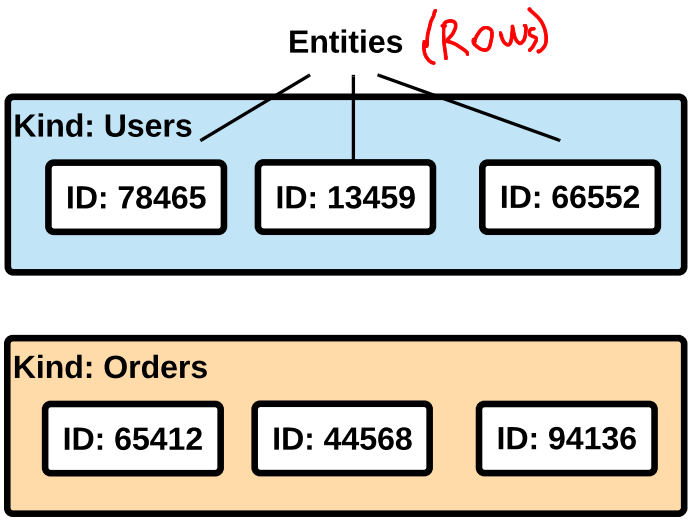
Queries and Indexing

Data Consistency

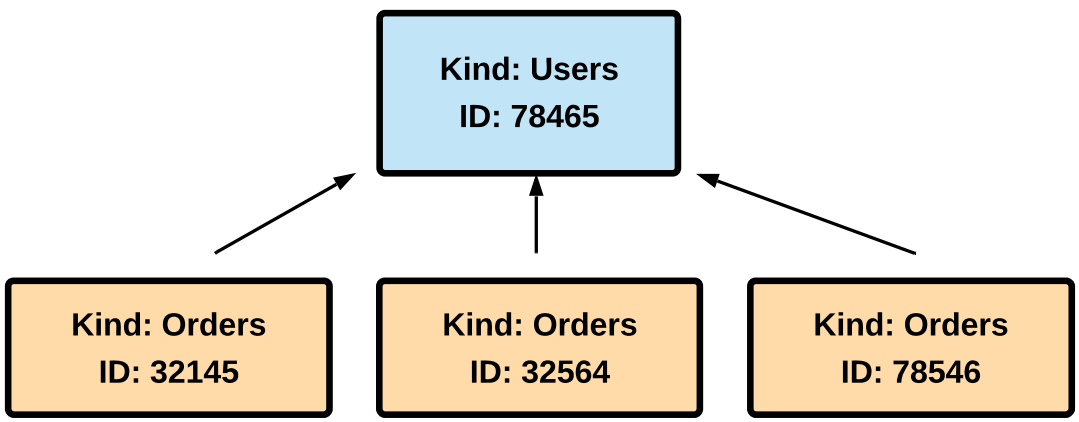
Data Organization

Previous

Simple Collections of Entities



Hierarchies (Entity Groups)



Return to Table of Contents

Choose a Lesson

- Cloud Datastore Overview
- Data Organization
- Queries and Indexing
- Data Consistency

Data Organization

Next

Short version:

- Entities grouped by kind (category)
- Entities can be hierarchical (nested)
- Each entity has one or more properties
- Properties have a value assigned

Key : Value stored

Concept	Relational Database	Datastore
Category of object	Table	Kind (category)
Single Object	Row	Entity
Individual data for an object	Column	Property
Unique ID for an object	Primary key	Key