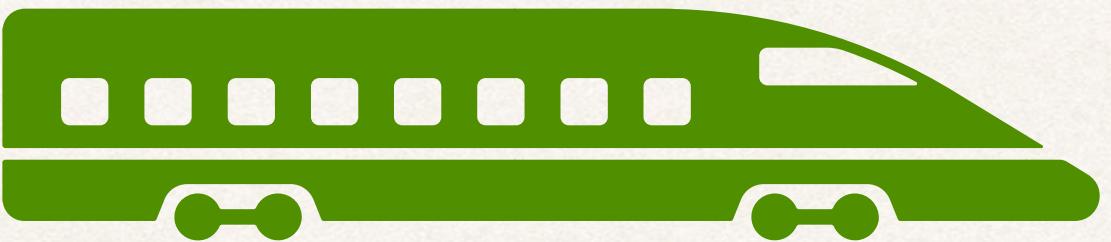
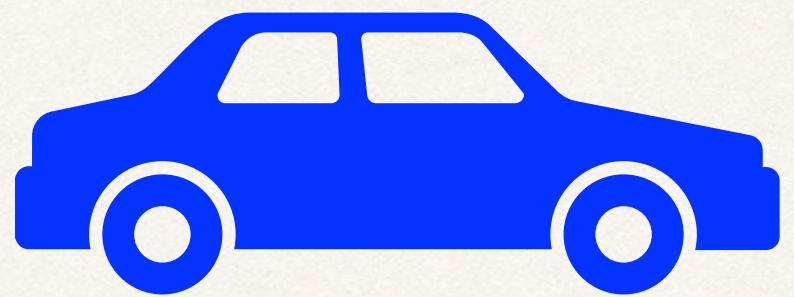


Each inheritance strategy has trade-offs

Each inheritance strategy has trade-offs



Guidance

Guidance

- Best performance with queries and write operations

Guidance

- Best performance with queries and write operations
 - **Single table**

Guidance

- Best performance with queries and write operations
 - **Single table**
- Good performance for queries of subclasses and data integrity

Guidance

- Best performance with queries and write operations
 - **Single table**
- Good performance for queries of subclasses and data integrity
 - **Table per class or Mapped Superclass**

Guidance

- Best performance with queries and write operations
 - **Single table**
- Good performance for queries of subclasses and data integrity
 - **Table per class or Mapped Superclass**
- Normalized database design and data integrity

Guidance

- Best performance with queries and write operations
 - **Single table**
- Good performance for queries of subclasses and data integrity
 - **Table per class or Mapped Superclass**
- Normalized database design and data integrity
 - **Joined table**

Guidance

- Best performance with queries
 - **Single table**
- Good performance for queries
 - **Table per class or Mapped**
- Normalized database design and data integrity
 - **Joined table**

Your mileage may vary

Depends on a number of factors:

- complexity of your inheritance tree
- number of rows in database
- database tuning
- application requirements

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | | | | |
| Performance: query concrete subclass | | | | |
| Performance: polymorphic query | | | | |
| Normalized database design | | | | |
| Data integrity (supports NOT NULL column constraints) | | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | | | | |
| Performance: query concrete subclass | | | | |
| Performance: polymorphic query | | | | |
| Normalized database design | | | | |
| Data integrity (supports NOT NULL column constraints) | | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | * | | | |
| Data integrity (supports NOT NULL column constraints) | * | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | * | | | |
| Data integrity (supports NOT NULL column constraints) | * | | | |

Fast because we are only performing operations on a single table

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Scale of 1 - 4:
1 star: the worst
4 stars: the best

Comparison

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Fast because we are only
performing query
on a single table

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Comparison

| | Table | Mapped Superclass |
|---|---------|-------------------|
| Performance: write operations - insert/update | 3 stars | |
| Performance: query concrete subclass | 4 stars | |
| Performance: polymorphic query | 4 stars | |
| Normalized database design | 1 star | |
| Data integrity (supports NOT NULL column constraints) | 1 star | |

For example:
Query for all User entities

Includes User subclasses
- Student, Instructor, etc ...

Scale of 1 - 4:
1 star: the worst
4 stars: the best

Scale of 1 - 4:
1 star: the worst
4 stars: the best

Comparison

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

For example:
Query for all User entities

Includes User subclasses
- Student, Instructor, etc ...

Fast because we are only
performing operations
on a single table

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Scale of 1 - 4:
1 star: the worst
4 stars: the best

Comparison

| | Single Table | Table Per Class | Table Per Subclass |
|---|--------------|-----------------|--------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★★ | ★★★★ |
| Performance: query concrete subclass | ★★★★ | ★★★★ | ★★★★ |
| Performance: polymorphic query | ★★★★ | ★★★★ | ★★★★ |
| Normalized database design | ★ | ★★★★ | ★★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★★ | ★★★★ |

Everything is in a single table

Not the ideal database design in regards to database normalization

Scale of 1 - 4:
1 star: the worst
4 stars: the best

Comparison

| | Single Table | Table Per Class | Table Per Subclass |
|---|--------------|-----------------|--------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★★ | ★★★★ |
| Performance: query concrete subclass | ★★★★ | ★★★★ | ★★★★ |
| Performance: polymorphic query | ★★★★ | ★★★★ | ★★★★ |
| Normalized database design | ★ | ★★★★ | ★★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★★ | ★★★★ |

Everything is in a single table

Not the ideal database design in regards to database normalization

Concept of Normalized Database Design

www.luv2code.com/normalization

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Joined Table | Table Per Class |
|---|--------------|--------------|-----------------|
| Performance: write operations - insert/update/delete | ★★★ | ★★★ | ★★★ |
| Performance: query concrete subclass | ★★★ | ★★★ | ★★★ |
| Performance: polymorphic query | ★★★ | ★★★ | ★★★ |
| Normalized database design | * | | |
| Data integrity (supports NOT NULL column constraints) | * | | |

Since table includes ALL fields of inheritance tree ...
the fields need to be nullable

For a given entity, we don't use all of the fields
(for example Instructor)

Can't apply
NOT NULL column constraints

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | * | | | |
| Data integrity (supports NOT NULL column constraints) | * | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | | | |
| Performance: query concrete subclass | ★★★★ | | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★ | ★★★★ | | |
| Performance: polymorphic query | ★★★★ | | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

For inserts, generally slower due to use of a sequence table.

ID generation with sequence tables in a high-volume multi-threaded environment is slower

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★ | ★★★★ | | |
| Performance: polymorphic query | ★★★★ | ★ | | |
| Normalized database design | ★ | | | |
| Data integrity (supports NOT NULL column constraints) | ★ | | | |

Fast because we are only performing query on a single table

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|---|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★ | Slow performance due to the need to join across multiple tables | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★ | <p>Normalized database design No empty fields</p> <p>Only have fields inherited and those of your subclass in the table</p> | |
| Performance: query concrete subclass | ★★★★ | ★ | | |
| Performance: polymorphic query | ★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★ | ★★★★ | | |
| Performance: polymorphic query | ★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Can apply
NOT NULL
column constraints
for data integrity

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★ | ★★★★ | | |
| Performance: polymorphic query | ★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|---|
| Performance: write operations - insert/update/delete | ★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★ | ★★★★ | ★ | |
| Performance: polymorphic query | ★★★★ | ★ | | <p>Slow performance</p> <p>To insert one entity, inserts into multiple tables based on primary key / foreign key</p> |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|--|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | | Slow performance because of multiple joins |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★ ★ ★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | Slow performance because of multiple joins | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|--|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | | Very good normalized database design |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | Inheritance relationships modeled using primary key and foreign key |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | | |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | ★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★ | |

Can apply
NOT NULL
column constraints
for data integrity

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | |
| Normalized database design | ★ | ★★★ | ★★★★★ | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | | |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Fast because we are only performing operations on a subclass table

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | | |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | | |

Fast because we are only performing query on a subclass table

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | | | | Mapped Superclass |
|---|------|--|------|-------------------|
| Performance: write operations - insert/update/delete | ★ | No direct support for polymorphic query since superclass is not an entity Need to manually create HQL which results in many joins | ★ | ★★★★ |
| Performance: query concrete subclass | ★★★★ | ★★★★ | ★ | ★★★★ |
| Performance: polymorphic query | ★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★ | ★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★☆ | | | ★★★★ |
| Performance: query concrete subclass | ★☆ | | | ★★★★ |
| Performance: polymorphic query | ★★☆☆ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★ | ★★★★ |

Good database design
No empty fields

Only have fields inherited and
those defined in subclass

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|---|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★ | ★★★★ | ★★★★ | ★★★★ |
| Performance: query concrete subclass | ★ | ★★★★ | ★★★★ | ★★★★ |
| Performance: polymorphic query | ★ | ★★★★ | ★★★★ | ★ |
| Normalized database design | ★ | ★★★★ | ★★★★ | ★★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★★ | ★★★★ | ★★★★ |

Can apply
NOT NULL
column constraints
for data integrity

Comparison

Scale of 1 - 4:
1 star: the worst
4 stars: the best

| | Single Table | Table per class | Joined table | Mapped Superclass |
|--|--------------|-----------------|--------------|-------------------|
| Performance: write operations - insert/update/delete | ★★★★★ | ★★★ | ★ | ★★★★★ |
| Performance: query concrete subclass | ★★★★★ | ★★★★★ | ★ | ★★★★★ |
| Performance: polymorphic query | ★★★★★ | ★ | ★ | ★ |
| Normalized database design | ★ | ★★★ | ★★★★★ | ★★★ |
| Data integrity (supports NOT NULL column constraints) | ★ | ★★★ | ★★★★★ | ★★★★★ |