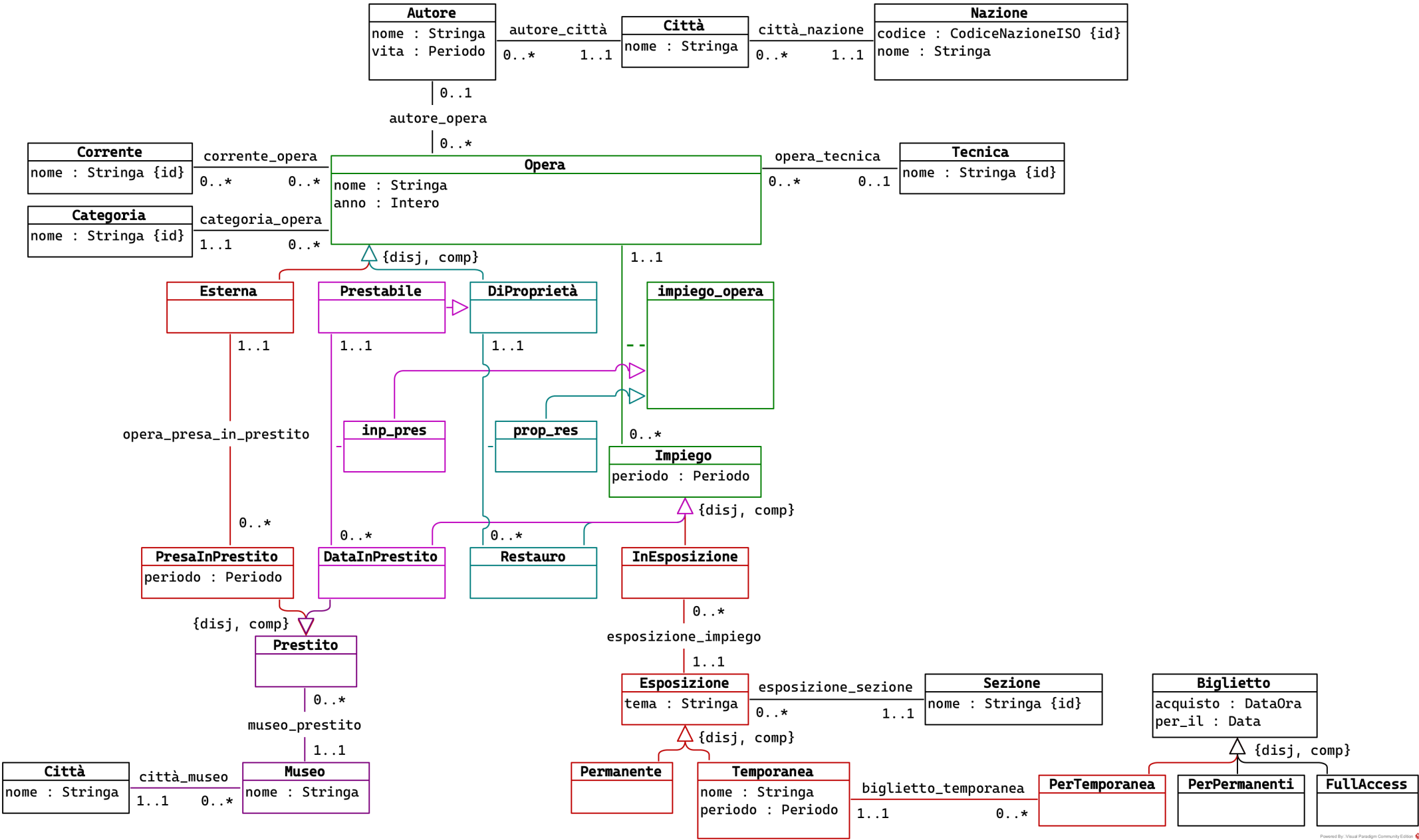


## Sommario

UML .....	3
Specifica dei tipi di dato .....	4
Specifica tipo <code>Periodo</code> .....	4
[V. <code>Periodo</code> . <code>inizio_prima_di_fine</code> ] .....	4
$\subseteq$ (p1: <code>Periodo</code> , p2: <code>Periodo</code> ): <code>Booleano</code> .....	4
$\in$ (t: <code>DataOra</code> , p2: <code>Periodo</code> ): <code>Booleano</code> .....	4
Specifica classe <code>Opera</code> .....	5
[V. <code>Opera</code> . <code>in_vita_autore</code> ] .....	5
[V. <code>Opera</code> . <code>no_impieghi_contemporanei</code> ] .....	5
[V. <code>Opera</code> . <code>impieghi_post_realizzazione</code> ] .....	5
Specifica classe <code>Biglietto</code> .....	5
[V. <code>Biglietto</code> . <code>acquistato_prima_di_utilizzo</code> ] .....	5
Specifica classe <code>PerTemporanea</code> .....	5
[V. <code>PerTemporanea</code> . <code>esposizione_valida</code> ] .....	5
Specifica classe <code>Esterna</code> .....	6
[V. <code>Esterna</code> . <code>opera_impiegata_durante_prestiti</code> ] .....	6
Specifica classe <code>Temporanea</code> .....	6
[V. <code>Temporanea</code> . <code>impieghi_in_periodo</code> ] .....	6
Use Case .....	7
Specifica use case <code>Gestione opere</code> .....	7
<code>opera</code> (	
nome: <code>Stringa</code> , anno: <code>Intero</code> , autore: <code>Autore</code> ,	
categoria: <code>Categoria</code> , corrente: <code>Corrente</code> [0..*], tecnica: <code>Tecnica</code> ,	
interna: <code>Booleano</code> , prestabile: <code>Booleano</code> [0..1]	
): <code>Opera</code> .....	7
<code>restauro</code> (opera: <code>DiProprietà</code> , periodo: <code>Periodo</code> ): <code>Restauro</code> .....	7
<code>presta</code> (opera: <code>Prestabile</code> , periodo: <code>Periodo</code> ): <code>DataInPrestito</code> .....	7
Specifica use case <code>Gestione esposizioni</code> .....	7
<code>permanente</code> (tema: <code>Stringa</code> , sezione: <code>Sezione</code> ): <code>Permanente</code> .....	7
<code>temporanea</code> (nome: <code>Stringa</code> tema: <code>Stringa</code> , sezione: <code>Sezione</code> , periodo: <code>Periodo</code> ):	
<code>Temporanea</code> .....	7
<code>in_esposizione</code> (opera: <code>Opera</code> , esposizione: <code>Esposizione</code> , periodo: <code>Periodo</code> ):	
<code>InEsposizione</code> .....	7
Specifica use case <code>Statistiche biglietti</code> .....	8
<code>vendite_per_tipologia</code> (data: <code>Data</code> ): ( <code>Intero</code> $\geq$ 0, <code>Intero</code> $\geq$ 0, <code>Intero</code> $\geq$ 0) .....	8
<code>media_vendite</code> (periodo: <code>Periodo</code> ): <code>Reale</code> $\geq$ 0 .....	8
<code>esposizioni_gettonate</code> (periodo: <code>Periodo</code> ): <code>Temporanea</code> [0..*] .....	8
Specifica use case <code>Informazioni opere</code> .....	9
<code>autori_per_corrente</code> (periodo: <code>Periodo</code> ): ( <code>Corrente</code> , <code>Intero</code> $\geq$ 0) [0..*] .....	9
Specifica use case <code>Informazioni restauri</code> .....	9
<code>opere_non_in_restaurato</code> (autore: <code>Autore</code> , periodo: <code>Periodo</code> ): <code>Opera</code> [0..*] .....	9
<code>opere_in_restaurato_ultima_esposizione</code> (): ( <code>Opera</code> , <code>Data</code> ) [0..*] .....	9
UML Ristrutturato (PostgreSQL) .....	10
Specifica ristrutturata dei tipi di dato .....	11
Vincoli aggiuntivi .....	11
[V. <code>Impiego</code> . <code>esposizione_sse_in_esposizione</code> ] .....	11
[V. <code>Impiego</code> . <code>restauro_sse_diproprietà</code> ] .....	11
[V. <code>Impiego</code> . <code>museo_sse_data_in_prestito</code> ] .....	11
[V. <code>Impiego</code> . <code>data_in_prestito_sse_prestabile</code> ] .....	11
[V. <code>Opera</code> . <code>prestabile_sse_di_proprietà</code> ] .....	11
[V. <code>Opera</code> . <code>presa_in_prestito_sse_di_esterna</code> ] .....	11
[V. <code>Biglietto</code> . <code>per_temporanea_sse_temporanea</code> ] .....	12

[V.Esposizione.disjoint]	12
[V.Esposizione.complete]	12
Vincoli ristrutturati	12
[V.Esterna.opera_impiegata_durante_prestiti]	12
[V.Temporanea.impieghi_in_periodo]	12
Schema relazionale	13
Trigger per ulteriori vincoli e permessi	16
[V.Opera.in_vita_autore]	16
[V.Opera.in_vita_autore]	16
[V.Opera.no_impieghi_contemporanei]	16
[V.Opera.impieghi_post_realizzazione]	17
[V.Opera.impieghi_post_realizzazione]	17
[V.PerTemporanea.esposizione_valida]	17
Implementazioni use case	18
vendite_per_tipologia(data: date): (IntegerGEZ, IntegerGEZ, IntegerGEZ)	18
media_vendite(periodo: Periodo): RealGEZ	18
esposizioni_gettonate(periodo: Periodo): Integer [0..*]	18
autori_per_corrente(periodo: Periodo): (varchar, IntegerGEZ) [0..*]	18
opere_non_in_restauro(autore: Integer, periodo: Periodo): Integer [0..*]	19
opere_in_restauro_ultima_esposizione(): (Integer, Data) [0..*]	19



## Specifica dei tipi di dato

`Periodo: (inizio: DataOra, fine: DataOra [0..1])`  
`CodiceNazioneISO: Stringa` secondo standard `ISO 3166-1 alpha-2`

## Specifica tipo `Periodo`

`[V.Periodo.inizio_prima_di_fine]`

$\forall p, i, f \text{ Periodo}(p) \wedge \text{inizio}(p, i) \wedge \text{fine}(p, f) \implies i \leq f$

`⊆ (p1: Periodo, p2: Periodo): Booleano`

**pre-condizioni**

**post-condizioni**

`result = True`  $\iff \exists i1, i2$   
 $\text{inizio}(p1, i1) \wedge \text{inizio}(p2, i2) \wedge i1 \leq i2 \wedge$   
 $\forall f1, f2 \text{ fine}(p1, f1) \wedge \text{fine}(p2, f2) \implies f1 \leq f2$

`∈ (t: DataOra, p2: Periodo): Booleano`

**pre-condizioni**

**post-condizioni**

`result = True`  $\iff \exists i \text{ inizio}(p, i) \wedge i \leq t \wedge (\forall f \text{ fine}(p, f) \implies t \leq f)$

## Specifica classe **Opera**

### [V.Opera.in\_vita\_autore]

$\forall$  autore, opera, anno, vita  
     $\text{autore\_opera}(\text{autore}, \text{opera}) \wedge \text{vita}(\text{autore}, \text{vita}) \wedge \text{anno}(\text{opera}, \text{anno}) \Rightarrow$   
         $\exists$  istante, data  $\text{data}(\text{istante}, \text{data}) \wedge \text{anno}(\text{data}, \text{anno}) \wedge \text{istante} \in \text{vita}$

### [V.Opera.no\_impieghi\_contemporanei]

$\forall$  opera, i1, i2, p1, p2  
     $\text{impiego\_opera}(\text{opera}, \text{i1}) \wedge$   
     $\text{impiego\_opera}(\text{opera}, \text{i2}) \wedge$   
     $\text{periodo}(\text{i1}, \text{p1}) \wedge$   
     $\text{periodo}(\text{i2}, \text{p2}) \Rightarrow$   
         $\neg \exists t \text{DataOra}(t) \wedge t \in \text{p1} \wedge t \in \text{p2}$

### [V.Opera.impieghi\_post\_realizzazione]

$\forall$  opera, impiego, anno, periodo, inizio  
     $\text{impiego\_opera}(\text{impiego}, \text{opera}) \wedge$   
     $\text{anno}(\text{opera}, \text{anno}) \wedge$   
     $\text{periodo}(\text{impiego}, \text{periodo}) \wedge$   
     $\text{inizio}(\text{periodo}, \text{inizio}) \Rightarrow$   
         $\exists$  istante, data  
             $\text{DataOra}(\text{istante}) \wedge$   
             $\text{data}(\text{istante}, \text{data}) \wedge$   
             $\text{anno}(\text{data}, \text{anno}) \wedge \text{istante} \leq \text{inizio}$

## Specifica classe **Biglietto**

### [V.Biglietto.acquistato\_prima\_di\_utilizzo]

$\forall$  biglietto, acquisto, utilizzo  
     $\text{Biglietto}(\text{biglietto}) \wedge$   
     $\text{acquisto}(\text{biglietto}, \text{acquisto}) \wedge$   
     $\text{per\_il}(\text{biglietto}, \text{utilizzo}) \Rightarrow$   
         $\exists$  istante  
             $\text{DataOra}(\text{istante}) \wedge \text{data}(\text{istante}, \text{utilizzo}) \wedge \text{acquisto} \leq \text{istante}$

## Specifica classe **PerTemporanea**

### [V.PerTemporanea.esposizione\_valida]

$\forall$  biglietto, esposizione, periodo, utilizzo  
     $\text{biglietto\_temporanea}(\text{biglietto}, \text{esposizione}) \wedge$   
     $\text{periodo}(\text{esposizione}, \text{periodo}) \wedge$   
     $\text{per\_il}(\text{biglietto}, \text{utilizzo}) \Rightarrow$   
         $\exists$  istante  $\text{DataOra}(\text{istante}) \wedge \text{data}(\text{istante}, \text{utilizzo}) \wedge \text{istante} \in \text{periodo}$

## Specifica classe **Eterna**

### [V.Eterna.opera\_impiegata\_durante\_prestiti]

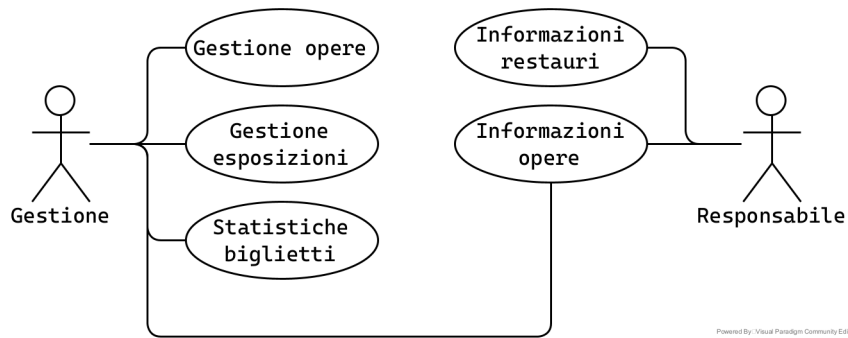
$\forall$  opera, impiego, periodo\_impiego  
Eterna(opera)  $\wedge$   
impiego\_opera(impiego, opera)  $\wedge$   
periodo(impiego, periodo\_impiego)  $\implies$   
 $\exists$  prestito, periodo\_prestito  
opera\_presa\_in\_prestito(opera, prestito)  $\wedge$   
periodo(prestito, periodo\_prestito)  $\wedge$   
periodo\_impiego  $\subseteq$  periodo\_prestito

## Specifica classe **Temporanea**

### [V.Temporanea.impieghi\_in\_periodo]

$\forall$  esposizione, periodo\_esposizione, impiego, periodo\_impiego  
Temporanea(esposizione)  $\wedge$   
esposizione\_impiego(esposizione, impiego)  $\wedge$   
periodo(esposizione, periodo\_esposizione)  $\wedge$   
periodo(impiego, periodo\_impiego)  $\implies$   
periodo\_impiego  $\subseteq$  periodo\_esposizione

## Use Case



## Specifica use case Gestione opere

```
opera(  
    nome: Stringa, anno: Intero, autore: Autore,  
    categoria: Categoria, corrente: Corrente [0..*], tecnica: Tecnica,  
    interna: Booleano, prestabile: Booleano [0..1]  
): Opera
```

```
restaura(opera: DiProprietà, periodo: Periodo): Restauro
```

```
presta(opera: Prestabile, periodo: Periodo): DataInPrestito
```

## Specifica use case Gestione esposizioni

```
permanente(tema: Stringa, sezione: Sezione): Permanente
```

```
temporanea(nome: Stringa tema: Stringa, sezione: Sezione, periodo: Periodo): Temporanea
```

```
in_esposizione(opera: Opera, esposizione: Esposizione, periodo: Periodo): InEsposizione
```

## Specifica use case Statistiche biglietti

**vendite\_per\_tipologia**(data: Data): (Intero  $\geq$  0, Intero  $\geq$  0, Intero  $\geq$  0)

pre-condizioni

post-condizioni

```
B = { biglietto |  $\exists$  acquisto acquisto(biglietto, acquisto)  $\wedge$  data(acquisto, data) }  
result = (  
  |{b | b  $\in$  B  $\wedge$  FullAccess(b)}|,  
  |{b | b  $\in$  B  $\wedge$  PerPermanenti(b)}|,  
  |{b | b  $\in$  B  $\wedge$  PerTemporanea(b)}|,  
)
```

**media\_vendite**(periodo: Periodo): Reale  $\geq$  0

pre-condizioni

post-condizioni

```
V = { biglietto |  $\exists$  acquisto  
  Biglietto(biglietto)  $\wedge$  acquisto(biglietto, acquisto)  $\wedge$  acquisto  $\in$  periodo  
}  
 $\exists$  inizio, fine  
  inizio(periodo, inizio)  $\wedge$  fine(periodo, fine)  $\implies$   
  result =  $\frac{|V|}{\text{fine} - \text{inizio} + 1}$ 
```

**esposizioni\_gettonate**(periodo: Periodo): Temporanea [0..\*]

pre-condizioni

post-condizioni

```
V = {(esposizione, vendite) |  $\exists$  periodo_esposizione  
  Temporanea(esposizione)  $\wedge$   
  periodo(esposizione, periodo_esposizione)  $\wedge$   
  vendite = { biglietto |  $\exists$  acquisto  
    acquisto(biglietto, acquisto)  $\wedge$   
    acquisto  $\in$  periodo  $\wedge$   
    (  
      FullAccess(biglietto)  $\wedge$   
       $\exists$  utilizzo, istante  
        per_il(biglietto, utilizzo)  $\wedge$   
        data(istante, utilizzo)  $\wedge$   
        istante  $\in$  periodo_esposizione  
    )  
  }  
  (  
    PerTemporanea(biglietto)  $\wedge$   
    biglietto_temporanea(biglietto, esposizione)  
  )  
}  
  
result = { esposizione |  $\exists$  valore (esposizione, valore)  $\in$  argmax  
  (e, v)  $\in$  V (v) }
```



## Specifica use case Informazioni opere

**autori\_per\_corrente**(periodo: Periodo): (Corrente, Intero  $\geq 0$ ) [0..\*]

pre-condizioni

post-condizioni

```
result = { (corrente, autori) |  
  Corrente(corrente)  $\wedge$   
  autori = |{ autore |  $\exists$  opera, anno, istante, data  
    corrente_opera(corrente, opera)  $\wedge$   
    autore_opera(autore, opera)  $\wedge$   
    anno(opera, anno)  $\wedge$   
    DataOra(istante)  $\wedge$   
    data(istante, data)  $\wedge$   
    anno(data, anno)  $\wedge$   
    istante  $\in$  periodo  
  }|  
}
```

## Specifica use case Informazioni restauri

**opere\_non\_in\_restauo**(autore: Autore, periodo: Periodo): Opera [0..\*]

pre-condizioni

post-condizioni

```
result = { opera |  
  autore_opera(autore, opera)  $\wedge$   
   $\neg \exists$  restauro, periodo_restauo, istante  
    prop_res(opera, restauro)  $\wedge$   
    periodo(restauro, periodo_restauo)  $\wedge$   
    DataOra(istante)  $\wedge$   
    istante  $\in$  periodo_restauo  $\wedge$   
    istante  $\in$  periodo  
}
```

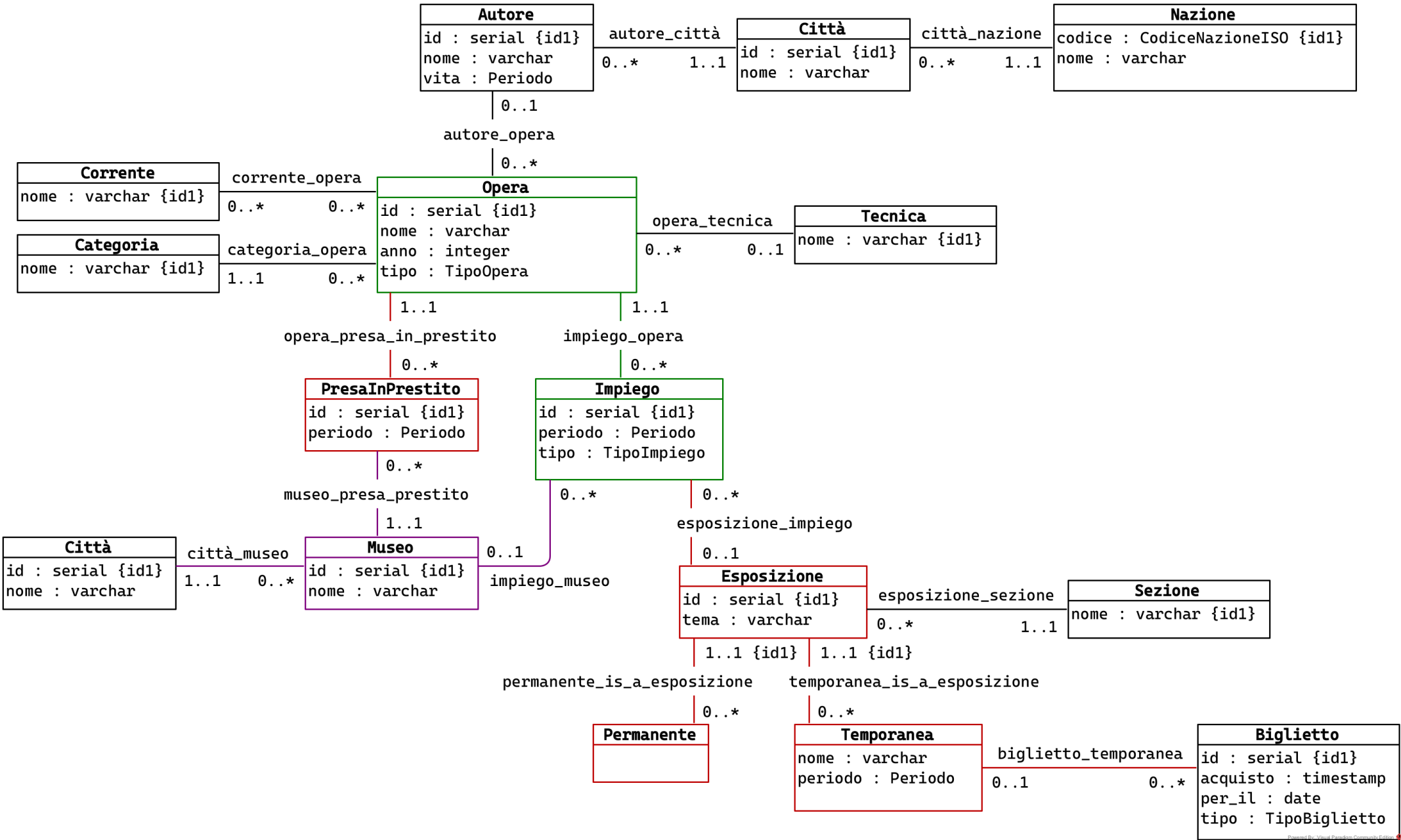
**opere\_in\_restauo\_ultima\_esposizione**(): (Opera, Data) [0..\*]

pre-condizioni

post-condizioni

```
0 = { (opera, data) |  $\exists$  i1, p1, f1  
  impiego_opera(i1, opera)  $\wedge$   
  InEsposizione(i1)  $\wedge$  periodo(i1, p1)  $\wedge$  fine(p1, f1)  $\wedge$   
  data(f1, data)  $\wedge$   
   $\neg$  exists i2, p2, f2  
    i1  $\neq$  i2  $\wedge$   
    impiego_opera(i2, opera)  $\wedge$   
    InEsposizione(i2)  $\wedge$  periodo(i2, p2)  $\wedge$  fine(p2, f2)  $\wedge$  f2  $\geq$  f1  
}  
  
result = { (opera, data) | (opera, data)  $\in$  0  $\wedge \exists$  restauro, periodo  
  Restauro(restauro)  $\wedge$   
  impiego_opera(opera, restauro)  $\wedge$   
  periodo(restauro, periodo)  $\wedge$   
  adesso  $\in$  periodo  
}
```

# UML Ristrutturato (PostgreSQL)



## Specifica ristrutturata dei tipi di dato

Periodo: (inizio: DataOra, fine: DataOra [0..1])  
CodiceNazioneISO: Stringa secondo standard ISO 3166-1 alpha 2  
TipoImpiego: { DataInPrestito, Restauro, InEsposizione }  
TipoOpera: { Esterna, DiProprietà }  
TipoBiglietto: { FullAccess, PerPermanenti, PerTemporanea }

```
CREATE TYPE raw_Periodo AS (inizio timestamp, fine timestamp);
CREATE DOMAIN Periodo AS raw_Periodo CHECK (
  (value).inizio IS NOT NULL AND (value).inizio <= (value).fine
);

CREATE DOMAIN CodiceNazioneISO AS CHAR(2) CHECK (value ~ '^[A-Z]{2}$');

CREATE TYPE TipoImpiego AS ENUM ( 'DataInPrestito', 'Restauro', 'InEsposizione' );
CREATE TYPE TipoOpera AS ENUM ( 'Esterna', 'DiProprietà' );
CREATE TYPE TipoBiglietto AS ENUM ( 'FullAccess', 'PerPermanenti', 'PerTemporanea' );

CREATE DOMAIN IntegerGEZ AS integer CHECK (value >= 0);
CREATE DOMAIN RealGEZ AS numeric CHECK (value >= 0);
```

## Vincoli aggiuntivi

### [V.Impiego.esposizione\_sse\_in\_esposizione]

$\forall$  impiego Impiego(impiego)  $\Rightarrow$   
    tipo(impiego, 'InEsposizione')  $\Leftrightarrow$   
     $\exists$  esposizione esposizione\_impiego(esposizione, impiego)

### [V.Impiego.restaurato\_sse\_diproprietà]

$\forall$  impiego Impiego(impiego)  $\Rightarrow$   
    tipo(impiego, 'Restauro')  $\Leftrightarrow$   
     $\exists$  opera impiego\_opera(impiego, opera)  $\wedge$  tipo(opera, 'DiProprietà')

### [V.Impiego.museo\_sse\_data\_in\_prestito]

$\forall$  impiego Impiego(impiego)  $\Rightarrow$   
    tipo(impiego, 'DataInPrestito')  $\Leftrightarrow \exists$  museo impiego\_museo(impiego, museo)

### [V.Impiego.data\_in\_prestito\_sse\_prestabile]

$\forall$  impiego Impiego(impiego)  $\Rightarrow$   
    tipo(impiego, 'DataInPrestito')  $\Leftrightarrow$   
     $\exists$  opera, prestabile  
        impiego\_opera(impiego, opera)  $\wedge$   
        prestabile(impiego, prestabile)  $\wedge$   
        prestabile = True

### [V.Opera.prestabile\_sse\_di\_proprietà]

$\forall$  opera Opera(opera)  $\Rightarrow$   
    tipo(opera, 'DiProprietà')  $\Leftrightarrow \exists$  prestabile prestabile(opera, prestabile)

### [V.Opera.presa\_in\_prestito\_sse\_di\_esterna]

$\forall$  opera Opera(opera)  $\Rightarrow$   
     $\exists$  prestito opera\_presa\_in\_prestito(opera, prestito)  $\Rightarrow$  tipo(opera, 'Esterna')

### [V.Biglietto.per\_temporanea\_sse\_temporanea]

$\forall$  biglietto `Biglietto`(biglietto)  $\implies$   
    `tipo`(biglietto, 'PerTemporanea')  $\iff$   
     $\exists$  temporanea `biglietto_temporanea`(biglietto, temporanea)

### [V.Esposizione.disjoint]

$\forall$  esposizione `Esposizione`(esposizione)  $\implies$   
     $\neg \exists$  permanente, temporanea  
        `permanente_is_a_esposizione`(permanente, esposizione)  $\wedge$   
        `temporanea_is_a_esposizione`(temporanea, esposizione)

### [V.Esposizione.complete]

$\forall$  esposizione `Esposizione`(esposizione)  $\implies$   
     $\exists$  permanente `permanente_is_a_esposizione`(permanente, esposizione)  $\vee$   
     $\exists$  temporanea `temporanea_is_a_esposizione`(temporanea, esposizione)

## Vincoli ristrutturati

### [V.Esterna.opera\_impiegata\_durante\_prestiti]

$\forall$  opera, impiego, periodo\_impiego  
    `tipo`(opera, 'Esterna')  $\wedge$   
    `impiego_opera`(impiego, opera)  $\wedge$   
    `periodo`(impiego, periodo\_impiego)  $\implies$   
         $\exists$  prestito, periodo\_prestito  
            `opera_presa_in_prestito`(opera, prestito)  $\wedge$   
            `periodo`(prestito, periodo\_prestito)  $\wedge$   
             $\text{periodo\_impiego} \subseteq \text{periodo\_prestito}$

### [V.Temporanea.impieghi\_in\_periodo]

$\forall$  esposizione, periodo\_esposizione, impiego, periodo\_impiego  
    `tipo`(esposizione, 'Temporanea')  $\wedge$   
    `esposizione_impiego`(esposizione, impiego)  $\wedge$   
    `periodo`(esposizione, periodo\_esposizione)  $\wedge$   
    `periodo`(impiego, periodo\_impiego)  $\implies$   
         $\text{periodo\_impiego} \subseteq \text{periodo\_esposizione}$

# Schema relazionale

```
CREATE TABLE Nazione (  
    codice CodiceNazioneISO NOT NULL,  
    nome varchar NOT NULL,  
    PRIMARY KEY (codice)  
);  
  
CREATE TABLE Citta (  
    id serial NOT NULL,  
    nome varchar NOT NULL,  
    nazione CodiceNazioneISO NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (nazione) REFERENCES Nazione(codice)  
);  
  
CREATE TABLE Autore (  
    id serial NOT NULL,  
    nome varchar NOT NULL,  
    vita Periodo NOT NULL,  
    citta integer,  
    PRIMARY KEY (id),  
    FOREIGN KEY (citta) REFERENCES Citta(id) DEFERRABLE  
);  
  
CREATE TABLE Categoria (  
    nome varchar NOT NULL,  
    PRIMARY KEY (nome)  
);  
  
CREATE TABLE Tecnica (  
    nome varchar NOT NULL,  
    PRIMARY KEY (nome)  
);  
  
CREATE TABLE Corrente (  
    nome varchar NOT NULL,  
    PRIMARY KEY (nome)  
);  
  
CREATE TABLE Opera (  
    id serial NOT NULL,  
    nome varchar NOT NULL,  
    anno integer NOT NULL,  
    tipo TipoOpera NOT NULL,  
    categoria varchar NOT NULL,  
    tecnica varchar,  
    autore integer,  
    PRIMARY KEY (id),  
    FOREIGN KEY (categoria) REFERENCES Categoria(nome) DEFERRABLE,  
    FOREIGN KEY (tecnica) REFERENCES Tecnica(nome) DEFERRABLE,  
    FOREIGN KEY (autore) REFERENCES Autore(id) DEFERRABLE,  
    CHECK (  
        (tipo = 'DiProprietà' AND prestabile IS NOT NULL) OR  
        (tipo <> 'DiProprietà' AND prestabile IS NULL)  
    )  
);  
  
CREATE TABLE corrente_opera (  
    opera integer NOT NULL,  
    corrente varchar NOT NULL,  
    PRIMARY KEY (opera, corrente),  
    FOREIGN KEY (opera) REFERENCES Opera(id) DEFERRABLE,  
    FOREIGN KEY (corrente) REFERENCES Corrente(nome) DEFERRABLE  
);
```

```

CREATE TABLE Museo (
    id serial NOT NULL,
    nome varchar NOT NULL,
    citta integer NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (citta) REFERENCES Citta(id) DEFERRABLE
);

CREATE TABLE PresaInPrestito (
    id serial NOT NULL,
    periodo Periodo NOT NULL,
    opera integer NOT NULL,
    museo integer NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (opera) REFERENCES Opera(id) DEFERRABLE,
    FOREIGN KEY (museo) REFERENCES Museo(id) DEFERRABLE
);

CREATE TABLE Sezione (
    nome varchar NOT NULL,
    PRIMARY KEY (nome)
);

CREATE TABLE Esposizione (
    id serial NOT NULL,
    tema varchar NOT NULL,
    sezione varchar NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (sezione) REFERENCES Sezione(nome) DEFERRABLE
);

CREATE TABLE Permanente (
    esposizione integer NOT NULL,
    PRIMARY KEY (esposizione),
    FOREIGN KEY (esposizione) REFERENCES Esposizione(id) DEFERRABLE
);

CREATE TABLE Temporanea (
    esposizione integer NOT NULL,
    nome varchar NOT NULL,
    periodo Periodo NOT NULL,
    PRIMARY KEY (esposizione),
    FOREIGN KEY (esposizione) REFERENCES Esposizione(id) DEFERRABLE
);

CREATE TABLE Biglietto (
    id serial NOT NULL,
    acquisto timestamp NOT NULL,
    per_il date NOT NULL,
    tipo TipoBiglietto NOT NULL,
    temporanea integer,
    PRIMARY KEY (id),
    FOREIGN KEY (temporanea) REFERENCES Temporanea(esposizione) DEFERRABLE,
    CHECK (
        (
            (tipo = 'PerTemporanea' AND temporanea IS NOT NULL) OR
            (tipo <> 'PerTemporanea' AND temporanea IS NULL)
        )
        AND date(acquisto) <= per_il
    )
);

```

```
CREATE TABLE Impiego (  
    id serial NOT NULL,  
    periodo Periodo NOT NULL,  
    tipo TipoImpiego NOT NULL,  
    opera integer NOT NULL,  
    museo integer,  
    esposizione integer,  
    PRIMARY KEY (id),  
    FOREIGN KEY (opera) REFERENCES Opera(id) DEFERRABLE,  
    FOREIGN KEY (museo) REFERENCES Museo(id) DEFERRABLE,  
    FOREIGN KEY (esposizione) REFERENCES Esposizione(id) DEFERRABLE,  
    CHECK (  
        (  
            (tipo = 'InEsposizione' AND esposizione IS NOT NULL) OR  
            (tipo <> 'InEsposizione' AND esposizione IS NULL)  
        )  
        AND  
        (  
            (tipo = 'DataInPrestito' AND museo IS NOT NULL) OR  
            (tipo <> 'DataInPrestito' AND museo IS NULL)  
        )  
    )  
);
```

# Trigger per ulteriori vincoli e permessi

## [V.Opera.in\_vita\_autore]

**operazioni intercettate** INSERT, UPDATE su Autore(vita)

**invocazione post**

**funzione**(old, new)

```
isError = EXISTS(
    SELECT *
    FROM Opera
    WHERE
        new.id = Opera.autore AND
        NOT Opera.anno BETWEEN
            EXTRACT(YEAR FROM (new.vita).inizio) AND
            EXTRACT(YEAR FROM COALESCE((new.vita).fine, 'infinity'))
)
if isError:
    raise Exception("...")
else:
    pass
```

## [V.Opera.in\_vita\_autore]

**operazioni intercettate** INSERT, UPDATE su Opera(autore, anno)

**invocazione post**

**funzione**(old, new)

```
isError = EXISTS(
    SELECT *
    FROM Autore
    WHERE
        new.autore = Autore.id AND
        NOT new.anno BETWEEN
            EXTRACT(YEAR FROM (Autore.vita).inizio) AND
            EXTRACT(YEAR FROM COALESCE((Autore.vita).fine, 'infinity'))
)
if isError:
    raise Exception("...")
else:
    pass
```

## [V.Opera.no\_impieghi\_contemporanei]

**operazioni intercettate** INSERT, UPDATE su Impiego(periodo, opera)

**invocazione post**

**funzione**(old, new)

```
isError = EXISTS(
    SELECT *
    FROM Impiego I
    WHERE
        I.opera = new.opera AND
        I.id <> new.id AND
        ((I.periodo).inizio, (I.periodo).fine) OVERLAPS
        ((new.periodo).inizio, (new.periodo).fine)
)
if isError:
    raise Exception("...")
else:
    pass
```



### [V.Opera.impieghi\_post\_realizzazione]

**operazioni intercettate** INSERT, UPDATE su Impiego(opera)

**invocazione post**

**funzione**(old, new)

```
isError = EXISTS(
    SELECT *
    FROM Opera
    WHERE
        Opera.id = new.opera AND
        EXTRACT(YEAR FROM (new.periodo).inizio) < Opera.anno
)
if isError:
    raise Exception("...")
else:
    pass
```

### [V.Opera.impieghi\_post\_realizzazione]

**operazioni intercettate** INSERT, UPDATE su Opera(anno)

**invocazione post**

**funzione**(old, new)

```
isError = EXISTS(
    SELECT *
    FROM Impiego
    WHERE
        new.id = Impiego.opera AND
        EXTRACT(YEAR FROM (Impiego.periodo).inizio) < new.anno
)
if isError:
    raise Exception("...")
else:
    pass
```

### [V.PerTemporanea.esposizione\_valida]

**operazioni intercettate** INSERT, UPDATE su Biglietto(temporanea, per\_il)

**invocazione post**

**funzione**(old, new)

```
isError = EXISTS(
    SELECT *
    FROM Temporanea T
    WHERE
        new.temporanea = T.id AND
        NOT new.per_il BETWEEN
            EXTRACT(DATE FROM (T.periodo).inizio) AND
            EXTRACT(DATE FROM COALESCE((T.periodo).fine, 'infinity'))
)
if isError:
    raise Exception("...")
else:
    pass
```

... ce ne sarebbero almeno altri 7, ma non li faccio tutti ...

## Implementazioni use case

**vendite\_per\_tipologia(data: date): (IntegerGEZ, IntegerGEZ, IntegerGEZ)**

```
SELECT B.tipo, COUNT(*)
FROM Biglietto B
WHERE date(B.acquisto) = 'data'
GROUP BY B.tipo
```

**media\_vendite(periodo: Periodo): RealGEZ**

```
WITH VenditeTotali AS (
  SELECT COUNT(*) AS COUNT
  FROM Biglietto B
  WHERE
    B.acquisto BETWEEN
      (periodo).inizio AND COALESCE((periodo).fine, 'infinity')
)
SELECT VenditeTotali.COUNT / ((periodo).inizio - COALESCE((periodo).fine, 'now'))
FROM VenditeTotali
```

**esposizioni\_gettonate(periodo: Periodo): Integer [0..\*]**

```
WITH EsposizioniVendite AS (
  SELECT T.esposizione AS id, COUNT(*) AS COUNT
  FROM Temporanea T, Biglietto B
  WHERE
    T.esposizione = B.temporanea OR
    (
      B.FullAccess AND
      B.per_il BETWEEN
        EXTRACT(DATE FROM (periodo).inizio) AND
        EXTRACT(DATE FROM COALESCE((periodo).fine, 'now'))
    )
  GROUP BY T.esposizione
),
MaxVendite AS (
  SELECT MAX(COUNT) AS MAX
  FROM EsposizioniVendite
)
SELECT id
FROM EsposizioniVendite
WHERE COUNT = MaxVendite.MAX
```

**autori\_per\_corrente(periodo: Periodo): (varchar, IntegerGEZ) [0..\*]**

```
SELECT C.nome, COUNT(DISTINCT A.id)
FROM
  (CORRENTE C LEFT JOIN Opera O ON C.nome = O.corrente)
  LEFT JOIN Artista A ON O.artista = A.id
WHERE
  O.anno BETWEEN
    EXTRACT(YEAR FROM (periodo).inizio) AND
    EXTRACT(YEAR FROM COALESCE((periodo).fine, 'now'))
GROUP BY C.nome
```

**opere\_non\_in\_restauro**(autore: Integer, periodo: Periodo): Integer [0..\*]

```
SELECT O.id
FROM Opera O
WHERE
  O.autore = 'autore' AND
  NOT EXISTS (
    SELECT *
    FROM Impiego I
    WHERE
      O.id = I.opera AND
      I.tipo = 'Restauro' AND
      ((I.periodo).inizio, (I.periodo).fine) OVERLAPS
      (('periodo').inizio, ('periodo').fine)
  )
```

**opere\_in\_restauro\_ultima\_esposizione**(): (Integer, Data) [0..\*]

```
WITH OpereRestauro AS (
  SELECT O.id
  FROM Opera O, Impiego I
  WHERE
    I.opera = O.id AND
    I.tipo = 'Restauro' AND
    current_timestamp BETWEEN
      (I.periodo).inizio AND
      COALESCE((I.periodo).fine, 'infinity')
)
SELECT O.id, MIN(DATE((I.periodo).fine)) AS date
FROM
  OpereRestauro O LEFT JOIN Impiego I ON I.opera = O.id
WHERE I.tipo = 'Esposizione'
GROUP BY O.id
```