

1. Vista sulle Mostre con Informazioni sui Curatori e il Numero di Recensioni

```
CREATE VIEW MostreCuratoriRecensioni AS

SELECT

    M.Nome AS NomeMostra,

    M.Prezzo,

    M.Descrizione,

    M.VotoMedio,

    C.Nome AS NomeCuratore,

    C.Cognome AS CognomeCuratore,

    COUNT(R.Timestamp) AS NumeroRecensioni

FROM

    Mostra M

LEFT JOIN

    Curatore C ON M.Curatore = C.CF

LEFT JOIN

    Recensione R ON M.Nome = R.Mostra

GROUP BY

    M.Nome, M.Prezzo, M.Descrizione, M.VotoMedio, C.Nome, C.Cognome;
```

2. Vista sui Restauratori e le loro Operazioni di Restauro

```
CREATE VIEW RestauratoriRestauri AS

SELECT

    R.CF,
```

```

        R.Nome,

        R.Cognome,

        R.Qualifica,

        COUNT(Rest.ID) AS NumeroRestauri,

        MAX(Rest.DataFine) AS UltimoRestauro,

        L.Nome AS NomeLaboratorio

FROM

        Restauratore R

LEFT JOIN

        Restauro Rest ON R.CF = Rest.RestauratoreID

LEFT JOIN

        Laboratorio L ON R.Laboratorio = L.Specializzazione

GROUP BY

        R.CF, R.Nome, R.Cognome, R.Qualifica, L.Nome;

```

3. Vista sui Prestiti con Informazioni sugli Enti e i Registrar Coinvolti

```

CREATE VIEW PrestitiDettagliati AS

SELECT

        P.ID,

        COALESCE(OI.Titolo, OE.Titolo) AS TitoloOpera,

        P.DataInizio,

        P.DataFine,

        E.Nome AS NomeEnte,

```

```

        E.Tipo AS TipoEnte,

        E.Indirizzo AS IndirizzoEnte,

        Reg.Nome AS NomeRegistrar,

        Reg.Cognome AS CognomeRegistrar

FROM

        Prestito P

LEFT JOIN

        OperaInterna OI ON P.ID_OperaInterna = OI.ID

LEFT JOIN

        OperaEsterna OE ON P.ID_OperaEsterna = OE.ID

LEFT JOIN

        Ente E ON P.Ente = E.Telefono

LEFT JOIN

        Registrar Reg ON P.Registrar = Reg.CF;

```

4. Vista sui Biglietti Venduti per Mostra con Informazioni sui Visitatori

```

CREATE VIEW BigliettiMostraVisitatore AS

SELECT

        B.NumeroSeriale,

        B.GiornoValidità,

        B.Audioguida,

        B.Sconto,

        B.PrezzoTot,

```

```
V.Nome AS NomeVisitatore,  
V.Cognome AS CognomeVisitatore,  
M.Nome AS NomeMostra,  
M.Prezzo AS PrezzoMostra  
FROM  
    Biglietto B  
LEFT JOIN  
    Visitatore V ON B.Visitatore = V.Email  
LEFT JOIN  
    Mostra M ON B.Mostra = M.Nome;
```

5. Vista sulle Partecipazioni agli Eventi da parte dei Curatori, Restauratori e Registrars

```
CREATE VIEW PartecipazioniEventi AS  
SELECT  
    E.Nome AS NomeEvento,  
    E.Data AS DataEvento,  
    E.Descrizione AS DescrizioneEvento,  
    Sala.ID AS IDSala,  
    Sala.Tipo AS TipoSala,  
    C.Nome AS NomeCuratore,  
    C.Cognome AS CognomeCuratore,  
    Res.Nome AS NomeRestauratore,  
    Res.Cognome AS CognomeRestauratore,
```

```

        Reg.Nome AS NomeRegistrar,

        Reg.Cognome AS CognomeRegistrar

FROM

        Evento E

LEFT JOIN

        Sala ON E.Sala = Sala.ID

LEFT JOIN

        PartecipazioneEventoCuratore PEC ON E.Sala = PEC.EventoSala AND
E.Data = PEC.EventoData

LEFT JOIN

        Curatore C ON PEC.Curatore = C.CF

LEFT JOIN

        PartecipazioneEventoRestauratore PER ON E.Sala = PER.EventoSala
AND E.Data = PER.EventoData

LEFT JOIN

        Restauratore Res ON PER.Restauratore = Res.CF

LEFT JOIN

        PartecipazioneEventoRegistrar PERG ON E.Sala = PERG.EventoSala
AND E.Data = PERG.EventoData

LEFT JOIN

        Registrar Reg ON PERG.Registrar = Reg.CF;

```

ecco una vista che calcola il voto medio delle mostre, tenendo conto delle recensioni associate a ciascuna mostra:

```
CREATE VIEW VotoMedioMostre AS

SELECT

    M.Nome AS NomeMostra,

    M.Descrizione,

    AVG(R.Voto) AS VotoMedio

FROM

    Mostra M

LEFT JOIN

    Recensione R ON M.Nome = R.Mostra

GROUP BY

    M.Nome, M.Descrizione;
```

Questa vista **VotoMedioMostre** restituisce il nome della mostra, la descrizione e il voto medio calcolato dalle recensioni per ciascuna mostra. Utilizza una join sinistra (**LEFT JOIN**) tra le tabelle **Mostra** e **Recensione**, in modo che tutte le mostre siano incluse nella vista, anche quelle che non hanno recensioni (in tal caso, il voto medio sarà **NULL**).

Ecco una vista che mostra il salario di ogni dipendente, indipendentemente dal loro ruolo (Curatore, Restauratore, Registrar, Direttore):

```
CREATE VIEW SalarioDipendenti AS
```

```
SELECT
```

```
    'Curatore' AS Ruolo,
```

```
    C.CF,
```

```
    C.Nome,
```

```
    C.Cognome,
```

```
    C.Retribuzione AS Salario
```

```
FROM
```

```
    Curatore C
```

```
UNION
```

```
SELECT
```

```
    'Restauratore' AS Ruolo,
```

```
    R.CF,
```

```
    R.Nome,
```

```
    R.Cognome,
```

```
    R.Retribuzione AS Salario
```

```
FROM
```

```
    Restauratore R
```

```
UNION
```

```
SELECT
```

```
    'Registrar' AS Ruolo,
```

```
    Reg.CF,
```

```
    Reg.Nome,
```

```
        Reg.Cognome,

        Reg.Retribuzione AS Salario

FROM

        Registrar Reg

UNION

SELECT

        'Direttore' AS Ruolo,

        D.CF,

        D.Nome,

        D.Cognome,

        D.Retribuzione AS Salario

FROM

        Direttore D;
```

Questa vista **SalarioDipendenti** utilizza l'operatore **UNION** per combinare i dati dalle tabelle **Curatore**, **Restauratore**, **Registrar** e **Direttore**. Ogni selezione specifica il ruolo del dipendente, il codice fiscale, il nome, il cognome e la retribuzione (salario) associata. Il risultato è una tabella unificata che mostra il salario di tutti i dipendenti, indipendentemente dal loro ruolo.