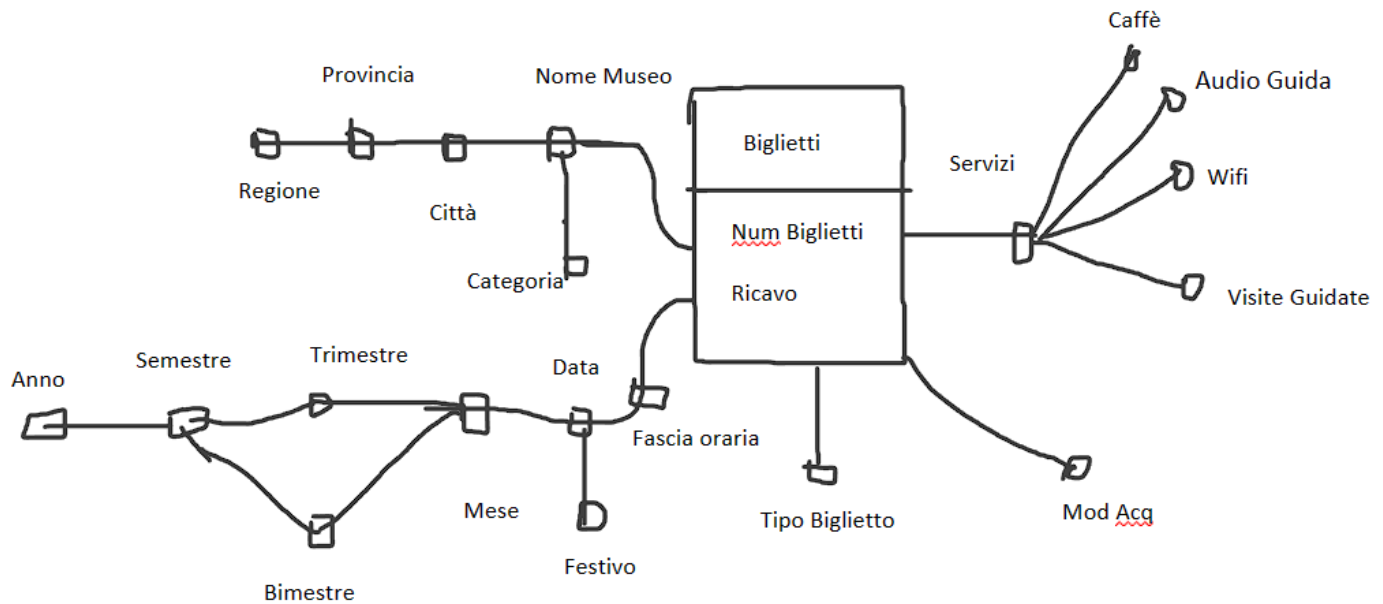


## PARTE 1

### SCHEMA CONCETTUALE



### SCHEMA LOGICO

SERVIZI(ID\_SERVIZIO, WI-FI, CAFFE, AUDIO\_GUIDE, VISITE\_GUIDATE, ...)  
INFO\_BIGLIETTO(ID\_INFO, TIPO\_BIGLIETTO, MOD\_ACQ)  
MUSEO(ID\_MUSEO, NOME\_MUSEO, CATEGORIA, CITTA, PROVINCIA, REGIONE)  
TEMPO(ID\_TEMPO, DATA, FESTIVO, FASCIA\_ORARIA, MESE, BIMESTRE, TRIMESTRE, SEMESTRE, ANNO)  
BIGLIETTI(ID\_TEMPO, ID\_MUSEO, ID\_INFO, ID\_SERVIZIO, NUM\_BIGLIETTI, RICAVO)

## PARTE 2

A)  
SELECT MESE, TIPO\_BIGLIETTO, SUM(RICAVO)/COUNT(DISTINCT DATA) AS MEDIA\_GIORNALIERA,  
SUM(SUM(RICAVO)) OVER (PARTITION BY ANNO, TIPO\_BILGIETTO  
ORDER BY MESE  
ROWS UNBOUNDED PREDCEING),  
100X SUM(NUM\_BIGLIETTI)/ SUM(SUM(NUM\_BIGLIETTI)) OVER (PARTITION BY MESE) AS PERCENTUALE  
FROM INFO\_BILGIETTO, BIGLIETTI, TEMPO  
WHERE INFO\_BIGLIETTO.ID\_INFO= BIGLIETTI.ID\_INFO AND TEMPO.ID\_TEMPO = BIGLIETTI.ID\_TEMPO  
GROUP BY MESE, TIPO\_BIGLIETTO

B)  
SELECT NOME\_MUSEO, TIPO\_BIGLIETTO, SUM(RICAVO) /SUM(NUM\_BIGLIETTI) AS MEDIA\_BIGLIETTO,  
100 X SUM(RICAVO)/SUM(SUM(RICAVO)) OVER (PARTITION BY TIPO\_BIGLIETTO, CATEGORIA),  
RANK () OVER (PARTITION BY TIPO\_BIGLIETTO, ORDER BY SUM(NUM\_BIGLIETTI) DESC)  
FROM MUSEO, INFO\_BIGLIETTO, BIGLIETTI, TEMPO  
WHERE ANNO=2021 AND INFO\_BIGLIETTO.ID\_INFO= BIGLIETTI.ID\_INFO AND TEMPO.ID\_TEMPO =  
BIGLIETTI.ID\_TEMPO AND MUSEO.ID\_MUSEO= BIGLIETTI.ID\_MUSEO  
GROUP BY NOME\_MUSEO, TIPO\_BIGLIETTO, CATEGORIA

### PARTE 3

1-SEL tipo di biglietto e per ogni semestre, SUM(RICAVO)/SUM()  
FROM BIGLIETTI, INFO\_BIGLIETTO, TEMPO  
GR TIPO\_B, SEMESTRE

2-SEL tipo di biglietto, MESE, SUM(VAL\_B) OVER (PARTITION BY ANNO  
ROWS UNBOUNDED PRECEDING)  
FROM BIGLIETTI, INFO\_BIGLIETTO, TEMPO  
GR TIPO\_B, MESE

3-SEL tipo di biglietto, MESE, SUM(NUM\_B), SUM(VAL\_B), SUM(VAL\_B)/ SUM(NUM\_B)  
FROM BIGLIETTI, INFO\_BIGLIETTO, TEMPO  
WHERE MOD\_ACQ=ONLINE  
GR TIPO\_B, SEMESTRE

4-SEL tipo di biglietto, MESE, SUM(NUM\_B), SUM(VAL\_B), SUM(VAL\_B)/ SUM(NUM\_B)  
FROM BIGLIETTI, INFO\_BIGLIETTO, TEMPO  
WHERE anno=2021.  
GR TIPO\_B, SEMESTRE

5-SEL tipo di biglietto, MESE, 100 X SUM(NUM\_BIGLI)/SUM(NUM\_B) OVER ( PARTITION BY MESE )  
FROM BIGLIETTI, INFO\_BIGLIETTO, TEMPO  
GR TIPO\_B, SEMESTRE

### 3.1 VISTA

```
CREATE MATERIALIZED VIEW VM1
BUILD IMMEDIATE AS
SELECT MESE, SEMESTRE, ANNO, TIPO_BIGLIETTO, MOD_ACQ, SUM(NUM_BIGLIETTI) AS TOT_BIGLIETTI,
SUM(RICAVO) AS TOT_RICAVO
FROM BIGLIETTI, TEMPO, INFO_BIGLIETTO
WHERE BIGLIETTI.ID_TEMPO=TEMPO.ID_TEMPO AND BIGLIETTI.ID_INFO = INFO_BIGLIETTO.ID_INFO
GROUP BY MESE, SEMESTRE, ANNO, TIPO_BIGLIETTO, MOD_ACQ
```

### 3.2 VISTA LOG

Le tabelle da tener traccia con i log per la vista materializzata sono:

BIGLIETTI : ID\_TEMPO ,ID\_INFO ,NUM\_BIGLIETTI,RICAVO  
INFO\_BIGLIETTO: ID\_INFO,TIPO\_BIGLIETTO,MOD\_ACQ  
TEMPO: ID\_TEMPO, MESE, SEMESTRE, ANNO

```
-CREATE MATERIALIZED VIEW LOG ON BIGLIETTI
WITH SEQUENCE,ROWID
(ID_TEMPO ,ID_INFO ,NUM_BIGLIETTI,RICAVO)
INCLUDING NEW VALUES
```

```
-CREATE MATERIALIZED VIEW LOG ON INFO_BIGLIETTO
WITH SEQUENCE,ROWID
(ID_INFO,TIPO_BIGLIETTO,MOD_ACQ)
INCLUDING NEW VALUES
```

```
-CREATE MATERIALIZED VIEW LOG ON TEMPO  
WITH SEQUENCE,ROWID  
(ID_TEMPO, MESE, SEMESTRE, ANNO)  
INCLUDING NEW VALUES
```

### 3.3

Le operazioni sulla base dati che causano un aggiornamento sulla materialized view sono le istruzioni di INSERT,UPDATE, DELETE nelle tabelle: BIGLIETTI, INFO\_BIGLIETTO, TEMPO

## PARTE 4

### 1.

```
CREATE TABLE VM1(  
MESE VARCHAR(20)CHECK (MESE IS NOT NULL),  
SEMESTRE VARCHAR(20)CHECK (SEMESTRE IS NOT NULL),  
ANNO VARCHAR(20) CHECK (ANNO IS NOT NULL),  
TIPO_BIGLIETTO VARCHAR(20) CHECK (TIPO_BIGLIETTO IS NOT NULL),  
MOD_ACQ VARCHAR(20) CHECK (MOD_ACQ IS NOT NULL),  
TOT_BIGLIETTI INTEGER,  
TOT_RICAVO INTEGER  
);
```

### 2.

```
INSERT INTO VM1( MESE,SEMESTRE, ANNO,TIPO_BIGLIETTO, MOD_ACQ, SUM(NUM_BIGLIETTI) AS TOT_BIGLIETTI,  
SUM(RICAVO) AS TOT_RICAVO)  
(SELECT MESE,SEMESTRE, ANNO,TIPO_BIGLIETTO, MOD_ACQ, SUM(NUM_BIGLIETTI), SUM(RICAVO)  
FROM BIGLIETTI, TEMPO, INFO_BIGLIETTO  
WHERE BIGLIETTI.ID_TEMPO=TEMPO.ID_TEMPO AND BIGLIETTI.ID_INFO = INFO_BIGLIETTO.ID_INFO  
GROUP BY MESE,SEMESTRE, ANNO,TIPO_BIGLIETTO, MOD_ACQ)
```

## 4.3 TRIGGER

```
CREATE OR REPLACE TRIGGER REFRESH-VIEW  
AFTER INSERT ON BIGLIETTI  
FOR EACH ROW
```

```
DECLARE
```

```
VARMESE VARCHAR(20) ,  
VARSEM VARCHAR(20) ,  
VARANNO VARCHAR(20) ,  
VAR_TIPO, VARCHAR(20) ,  
VAR_MOD VARCHAR(20) ,  
N INT;
```

```
BEGIN
```

```
SELECT MESE, SEMESTRE, ANNO  
INTO VARMESE,VARSEM,VARANNO  
FROM TEMPO  
WHERE ID_TEMPO=NEW.ID_TEMPO
```

```
SELCT TIPO_BIGLIETTO, MOD_ACQ  
INTO VAR_TIPO,VAR_MOD  
FROM INFO_BIGLIETTO  
WHERE ID_INFO=NEW.ID_INFO
```

```
SELECT COUNT(*) INTO N  
FORM VM1  
WHERE MESE=VARMESE AND TIPO_BIGLIETTO= VAR_TIPO AND MOD_ACQ=VAR_MOD
```

```
IF (N>0) THEN  
    UPDATE VM1  
    SET TOT_BIGLIETTI = TOT_BIGLIETTI +:NEW.NUM_BIGLIETTI,  
    TOT_RICAVO = TOT_RICAVO +:NEW.RICAVO  
    WHERE MESE=VARMESE AND TIPO_BIGLIETTO= VAR_TIPO AND MOD_ACQ=VAR_MOD
```

```
ELSE  
    INSERT INTO VM1 (MESE,SEMESTRE, ANNO,TIPO_BIGLIETTO, MOD_ACQUI, TOT_BIGLIETTI,  
    TOT_RICAVO)  
    VALUES( MESE,SEMESTRE, ANNO,TIPO_BIGLIETTO, MOD_ACQ,:NEW.NUM_BIGLIETTI,  
    :NEW.RICAVO )
```

```
END IF
```

```
END
```

#### 4.4

L'operazione sulla base dati che causano l'attivazione del trigger è l'istruzione di INSERT