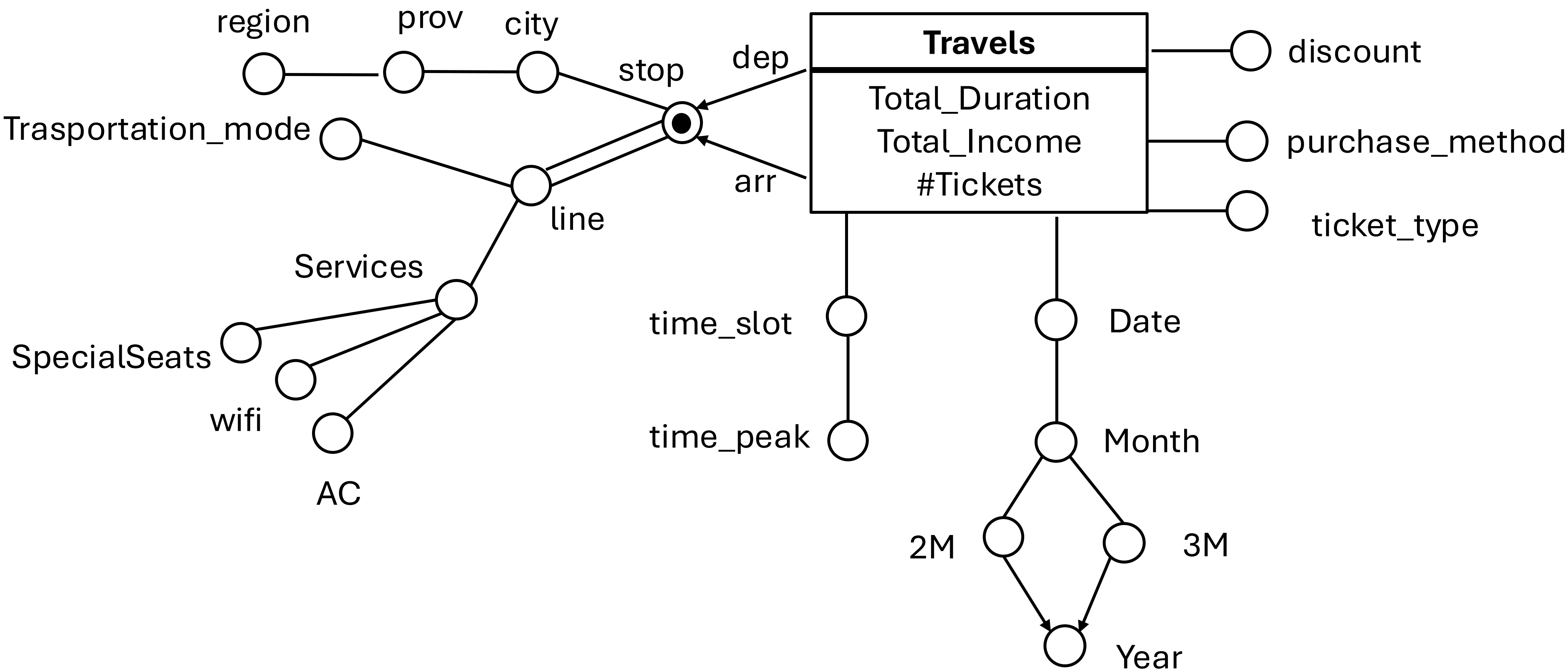
Homework 1

Conceptual design - sol 2



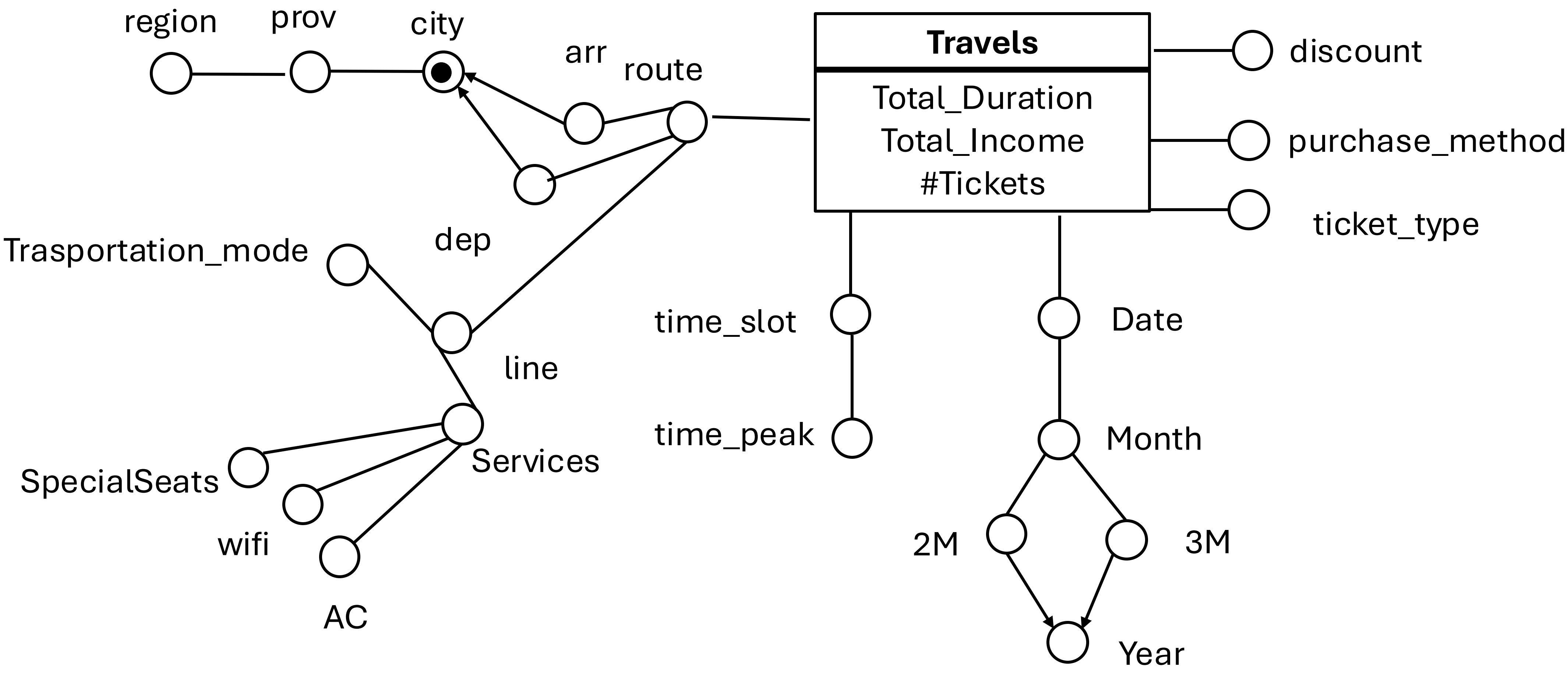
Logical design – sol 1

- STOP(CodST, Stop, City, Province, Region)
- LINE (CodL, Line, Mode, wifi, AC, SpecialSeats)
- STOP_LINE(CodST, CodL)
- TIME(CodT, Date, Month, 2M, 3M, Year)
- SLOT(CodS, Time_slot, Peak)
- TRAVELS(CodT, CodS, DepCodST, ArrCodST, Puchase_method, Discount, Ticket_type, Total_duration, #Tickets, Total_Income)

Alternative solution

- STOP(CodST, Stop, City, Province, Region)
- LINE (CodL, Line, Mode, wifi, AC, SpecialSeats)
- STOP_LINE(CodST, CodL)
- TIME(CodT, Date, Month, 2M, 3M, Year)
- SLOT(CodS, Time_slot, Peak)
- JUNK (CodJ, Puchase_method, Discount, Ticket_type)
- TRAVELS(CodT, CodS, DepCodST, ArrCodST, CodJ, Total_duration, #Tickets, Total_Income)

Conceptual design – sol 2



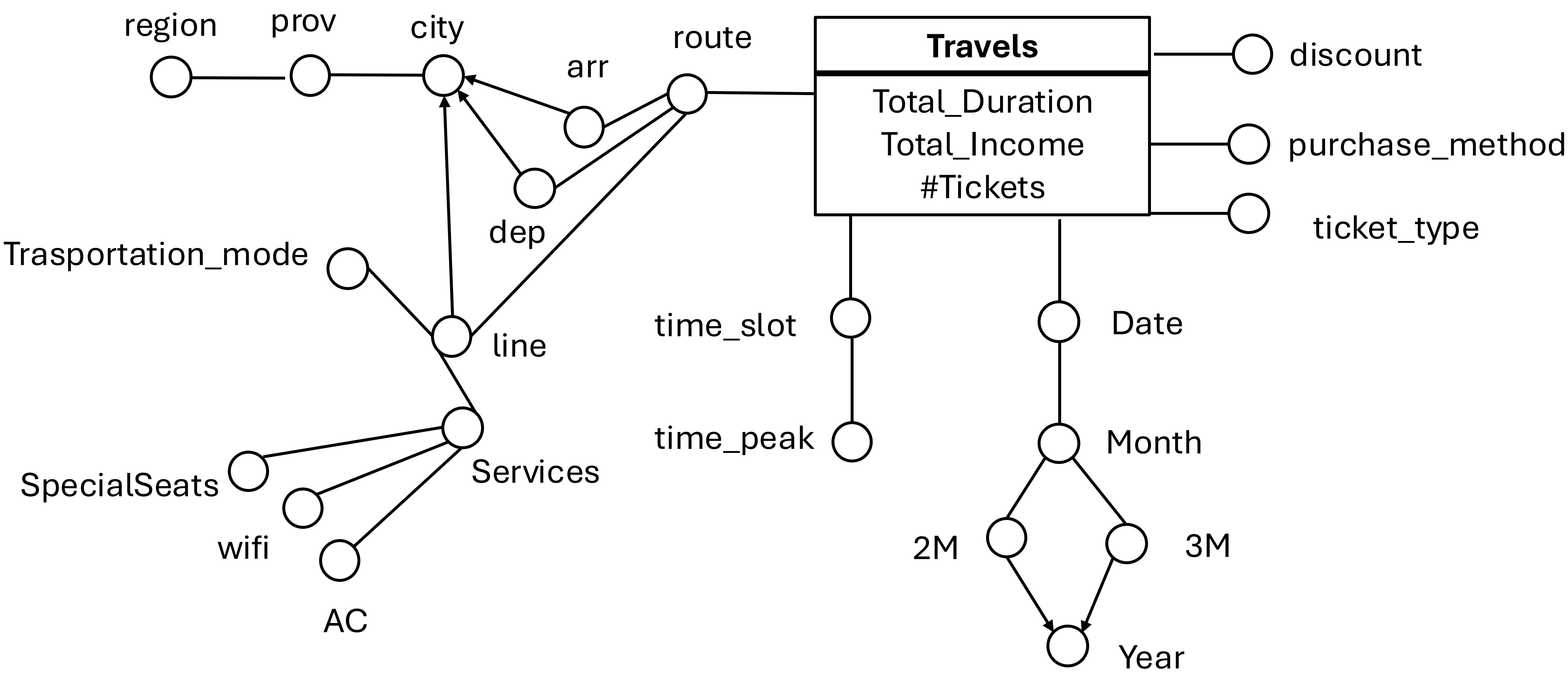
Logical design – sol 2

- LOCATION(CodL, City, Prov, Region)
- ROUTE(CodR, Route, Arr, Dep, ArrCodL, DepCodL, Line, Mode, wifi, AC, SpecialSeats)
- TIME(CodT, Date, Month, 2M, 3M, Year)
- SLOT(CodS, Time_slot, Peak)
- TRAVELS(CodT, CodS, CodR, Puchase_method, Discount, Ticket_type, Total_duration, #Tickets, Total_Income)

Alternative solution

- LOCATION(CodL, City, Prov, Region)
- ROUTE(CodR, Route, Arr, Dep, ArrCodL, DepCodL, Line, Mode, wifi, AC, SpecialSeats)
- TIME(CodT, Date, Month, 2M, 3M, Year)
- SLOT(CodS, Time_slot, Peak)
- JUNK (CodJ, Puchase_method, Discount, Ticket_type)
- TRAVELS(CodT, CodS, CodR, CodJ, Total_duration, #Tickets, Total_Income)

Conceptual design – sol 3



Logical design – sol 3

- ROUTE(CodR, Route, Arr, Dep, Line, Mode, wifi, AC, SpecialSeats, City, Prov, Region)
- TIME(CodT, Date, Month, 2M, 3M, Year)
- SLOT(CodS, Time_slot, Peak)
- TRAVELS(CodT, CodS, CodR, Puchase_method, Discount, Ticket_type, Total_duration, #Tickets, Total_Income)

Alternative solution

- ROUTE(CodR, Route, Arr, Dep, Line, Mode, wifi, AC, SpecialSeats, City, Prov, Region)
- TIME(CodT, Date, Month, 2M, 3M, Year)
- SLOT(CodS, Time_slot, Peak)
- JUNK (CodJ, Puchase_method, Discount, Ticket_type)
- TRAVELS(CodT, CodS, CodR, CodJ, Total_duration, #Tickets, Total_Income)

Query 1

Separately for each transportation mode and for each month, analyze:

- the average daily number of tickets,
- the cumulative number of tickets since the beginning of the year, and
- the percentage of tickets used for each transportation mode compared to the total number of tickets in that month.

```
SELECT

mode, month,

SUM(numtickets)/COUNT(distinct date),

SUM(SUM(numtickets)) over (PARTITION BY Mode, Year ORDER BY month ROWS UNBOUNDED PRECIDING),

100*SUM(numtickets)/SUM(SUM(numtickets)) over (PARTITION BY month)

FROM TRAVELS TR, ROUTE R, TIME T

WHERE TR.CodT = T.CodT and TR.CodR = R.CodR

GROUP BY mode, month, year
```

Query 2

Considering travels from 2022, separately for each line and city, we analyze:

- the average travel duration,
- the total revenues generated by that city,
- the percentage of total revenue contributed by each line to the city's total, and
- assign a rank to each line within its city based on total line revenues generated in descending order

```
SELECT
city, line,
SUM(Total_duration)/SUM(#Tickets),
SUM(SUM(Total_Income)) over (PARTITION BY City)
100*SUM(Total_Income)/SUM(SUM(Total_Income))
over (PARTITION BY City),
RANK() over (PARTITION BY City ORDER BY
SUM(Total_Income) desc)
FROM TRAVELS TR, ROUTE R, TIME T
WHERE TR.CodT = T.CodT and TR.CodR = R.CodR AND
year >= 2022
GROUP BY city, line
```

Materialized view

- Query 1
  - SELECT SUM(NumTickets), COUNT(DISTINCT Date)
  - GB: Mode, Month
- Query 2
  - SELECT SUM(NumTickets)
  - GB: Mode, Month, Year
- Query 3
  - SELECT SUM(NumTickets), SUM(Tot_Income)
  - GB: Mode, Month
- Query 4
  - SELECT SUM(NumTickets), SUM(Tot_Income)
  - WHERE Year=2024
  - GB: Mode, Month
- Query 5
  - SELECT SUM(NumTickets)
  - GB: Mode, Month

Materialized view – sol1

```
CREATE MATERIALIZED VIEW MV1
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS
SELECT Mode, Month, Year,
SUM(#Tickets) AS TotTickets,
SUM(Tot_Income) AS TotRevenue,
COUNT (DISTINCT Date)
FROM TRAVELS TR, ROUTE R, TIME T
WHERE TR.CodT = T.CodT and TR.CodR = R.CodR
GROUP BY Mode, Month, Year
```

Materialized view – sol2 with log

```
CREATE MATERIALIZED VIEW MV1
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS
SELECT Mode, Month, Year,  Date
SUM(#Tickets) AS TotTickets,
SUM(Tot_Income) AS TotRevenue,
FROM TRAVELS TR, ROUTE R, TIME T
WHERE TR.CodT = T.CodT and TR.CodR = R.CodR
GROUP BY Mode, Month, Year, Date
```

-
  ```
  CREATE MATERIALIZED VIEW LOG ON ROUTE
  WITH SEQUENCE, ROWID
  (CodR, Mode)
  INCLUDING NEW VALUES;
  ```

-
  ```
  CREATE MATERIALIZED VIEW LOG ON TIME
  WITH SEQUENCE, ROWID
  (CodT, Date, Month, Year)
  INCLUDING NEW VALUES;
  ```

-
  ```
  CREATE MATERIALIZED VIEW LOG ON TRAVELS
  WITH SEQUENCE, ROWID
  (CodT, CodR, NumTickets, Tot_Income)
  INCLUDING NEW VALUES;
  ```

Create materialized view table

```
CREATE TABLE VM1 (
   Mode VARCHAR (50) CHECK (Mode IS NOT
NULL),
   Date DATE  CHECK (Month IS NOT NULL)
   Month DATE  CHECK (Month IS NOT NULL)
   Year DATE  CHECK (Year IS NOT NULL),
   TotTickets INTEGER,
   TotRevenue DOUBLE PRECISION)
```

```
INSERT INTO VM1 (Mode, Date, Month, Year,
TotTickets, TotRevenue)
  (SELECT Mode, Date, Month, Year,
   SUM(NumTickets), SUM(Revenue)
   FROM TRAVELS TR, ROUTE R, TIME T
   WHERE TR.CodT = T.CodT AND
   TR.CodR = R.CodR
   GROUP BY Mode, Date, Month, Year );
```

Trigger – part 1

```
CREATE OR REPLACE TRIGGER Trigger1
 AFTER INSERT ON TRAVELS
 FOR EACH ROW
 DECLARE
 VarTranspMode VARCHAR (50);
 VarDate, VarMonth, VarYear DATE;
 N INTEGER ;

 BEGIN
            SELECT Mode INTO VarTranspMode
            FROM ROUTE
            WHERE CODR = :NEW.CODR;
```

```
SELECT Date, Month, Year INTO
VarDate, VarMonth, VarYear
FROM TIME
WHERE CODET = :NEW.CODT;

SELECT COUNT (*) INTO N
FROM VM1
WHERE Mode = VarTranspMode
AND Date = VarDate;
```

Trigger – part 2

```
    IF (N > 0) THEN
        UPDATE MV1
        SET TotTickets += :NEW.NumTickets,
         TotRevenue += :NEW.Tot_Income
        WHERE Mode = VarTranspMode
        AND Date = VarDate;

    ELSE
        INSERT INTO MV1 (Mode, Date, Month, Year, NumTickets, Revenue)
        VALUES (VarTranspMode, VarDate, VarMonth, VarYear, :NEW.NumTickets,
        :NEW.Revenue);
    END IF;
END;
```