Problem 1. A sale was done during a travel if the store of the sale was not in the city of residence of the customer. It is required to produce a CSV with a row for every customer and her/his percentage of sales done during travel over the total sales of the customer.

SQL

Abbiamo assunto che il tipo di dato richiesto nel risultato sia il conteggio del numero degli acquisti del customer, pertanto la prima tabella necessaria è **sales_fact**. Inoltre è necessario conoscere per ogni acquisto: la città di residenza del customer e la città dello store dove è stato effettuato. Quindi le altre due tabelle necessarie sono rispettivamente **customer** e **store**. Le suddette tabelle vanno messe in relazione mediante join utilizzando le foreign key di **sales_fact**: **customer_id** e **store_id**, e le rispettive chiavi della tabella **customer** (**customer_id**) e della tabella **store** (**store_id**).

```
14  from sales_fact sf, customer c, store s
15  where sf.customer_id = c.customer_id and sf.store_id = s.store_id
```

A questo punto è necessario raggruppare per **customer_id** in quanto la granularità dell'informazione richiesta è il customer.

```
16  group by sf.customer_id
```

Per ogni gruppo generato:

- prendiamo il **customer_id**
- contiamo il numero degli acquisti fatti dal customer in città <u>diverse</u> dalla propria città di residenza (**tot_travel_sales**)
- contiamo il numero totale di <u>tutti</u> gli acquisti del customer (**tot_sales**)
- calcoliamo il rapporto tra **tot_travel_sales** e **tot_sales**, espresso in percentuale.

```
5  select
6      sf.customer_id,
7      count(case when s.store_city <> c.city then 1 else null end) as tot_travel_sales,
8      count(sf.fact_id) as tot_sales,
9      cast(
10         (
11             1.0 * count(case when s.store_city <> c.city then 1 else null end) / count(sf.fact_id)
12         ) * 100 as numeric(5,2)
13     ) as percent_of_travel_tot_sales
```

SSIS

1. Crea la <u>sorgente OLE DB</u> a partire dalla tabella **sales_fact(fact_id, customer_id, store_id, store_sales)**;
2. Duplica la sorgente **sales_fact** tramite <u>multicast</u>: **multicast_1** e **multicast_2**
    a. **multicast_1**

   i. per ogni acquisto (o vendita) cerca **store_city** nella tabella **store**, e **customer_city** nella tabella **customer**;
   ii. seleziona solo gli acquisti fatti dai customer in città diverse dalla propria città di residenza;
   iii. raggruppa per **customer_id** e conta il numero degli acquisti per customer (**tot_travel_sales**);
2. b. **multicast_2**
   i. raggruppa per **customer_id** e conta il numero totale degli acquisti per customer (**tot_sales**);
3. Join delle due tabelle risultanti da **multicast_1** (**customer_id**, **tot_travel_sales**) e **multicast_2** (**customer_id**, **tot_sales**);
4. Aggiungi la colonna **percent_of_travel_tot_sales** come rapporto tra **tot_travel_sales** e **tot_sales** espresso in percentuale con la colonna derivata;
5. Scrivi su file output1.csv la tabella: (**customer_id, tot_travel_sales, tot_sales, percent_of_travel_tot_sales**) con destinazione file flat.

Problem 3. The quarter value QV of a customer id C at the date id T is the sum of revenues minus costs of the products that the customer C buys in the quarter of T. It is required to produce a CSV le with three columns: customer id, time it, QV.

SQL

Abbiamo interpretato la risoluzione della query con l'elaborazione di un output che fornisse il QV (cioè il ricavo comportato da ogni singolo cliente per ogni trimestre), per ogni cliente e per ogni time id.
Per prima cosa abbiamo effettuato la join fra la tabella **sales_fact** e **time_by_day** ed una group by per **customer_id**, **time_id**, **quarter**, **the_year**.

```
from sales_fact sf JOIN time_by_day t ON sf.time_id=t.time_id
group by sf.customer_id, sf.time_id,t.quarter, t.the_year
```

Per ottenere il valore di QV abbiamo fatto la differenza fra la somma di **store_sales** e la somma di **store_cost**, la cui somma abbiamo poi raggruppato con una partition by per cliente, trimestre e anno.

```
sum(sum(sf.store_sales - sf.store_cost)) OVER
(PARTITION BY sf.customer_id, t.quarter,t.the_year) AS QV
```

Abbiamo poi ordinato i risultati per cliente e trimestre.

```
order by customer_id, t.quarter ;
```

Nella select, per maggiore leggibilità, abbiamo aggiunto le colonne dell'anno e del trimestre.

SSIS

1. Creo una sorgente OLE BD in cui importo la tabella **sales_fact** da Foodmart, da cui importo le colonne di **time_id**, **customer_id**, **store_sales**, **store_cost**.
2. Con un operatore di Look up, unisco alla taballa **sales_fact** la tabella **time_by_day**, da cui importo le colonne **quarter** e **the_year**.
3. Con il Multicast sdoppio il flusso di dati.
4. Nel primo flusso effettuo un'operazione di group by su **time_id**, **customer_id**, **the_year**, **quarter** e poi ordino per cliente, trimestre e anno.
5. Nel secondo flusso, nel nodo di aggregazione effettuo le somme di **store_sales** e **store_cost** e il group by per **customer_id**, **quarter**, **the_year**.
6. Nella colonna derivata effettuo la differenza fra **store_sales** e **store_cost** e poi ordino per cliente, trimestre e anno.
7. Con la Merge join riunisco i due flussi facendo il mapping con **customer_id**, **quarter** e **the_year**.
8. Creo il file csv con destinazione file flat.

Problem 4 A product id is female-specific if the number of distinct women customers who bought the product is at least 1.5 times the number of distinct male customers who bought it. It is required to produce a CSV with all female-specific product id's.

SQL

L'interpretazione data a questa consegna è stata che quest'ultima ci chiedesse di riportare prodotti definiti "female-specific", ossia prodotti acquistati almeno una volta e mezzo in più da clienti donne rispetto a clienti uomini.

I dati che ci servivano erano i seguenti: **product_id** (e possibilmente nome del prodotto per identificarlo meglio), **customer_id**, **gender** dei customer (per dividere uomini e donne).

Questi dati si trovano in due tabelle del database FoodMart: **Sales_fact, Customer e Product**. **Sales_fact** contiene **customer_id** e **product_id**, ma per risalire al sesso del cliente e al nome del prodotto abbiamo effettuato una join tra tabella **sales_fact** sia con la tabella **Customer**, usando la foreign key **customer_id**, sia con la tabella **Product**, tramite la foreign key **product_id.**

```
from sales_fact s join customer c on c.customer_id = s.customer_id
join product p on p.product_id = s.product_id
```

Raggruppiamo i dati per product_id e per product_name:

```
group by s.product_id, p.product_name
```

e procediamo a contare quanti clienti donne e uomini hanno acquistato i prodotti, ricordandoci di contare ogni cliente solo una volta (specificando quindi: *distinct*).

```
    count (distinct case when c.gender = 'f' then c.customer_id else null end) as FemaleCount,
    count (distinct case when c.gender = 'm' then c.customer_id else null end) as MaleCount
```

Il risultato ottenuto nelle query precedenti appartiene a una vista temporanea, che ci servirà per selezionare solo i prodotti acquistati una volta e mezzo da donne rispetto a uomini.

```
from Gender_Products_Count g
where g.FemaleCount >= g.MaleCount*1.5
```

Selezioniamo poi le voci che desideriamo appaiano nella tabella finale: **product_id, product_name select product_id, product_name.**

```
select g.product_id, g.product_name
```

SSIS

1. Crea la sorgente OLE DB a partire dalla tabella **sales_fact(product_id, customer_id, store_sales)**;
2. Unisco la tabella **sales_fact** con la tabella **customer** tramite la funzione ricerca, unendo le tabelle tramite la chiave **customer_id**
3. Divido i **customer donne e uomini** con la funzione multicast sul valore gender
4. Con una funzione di aggregazione:

    I. raggruppando per **product_id** con la funzione group by
    II. conto **i clienti donne e uomini** con la funzione count
    III. sommo quindi le vendite **store_sales,** che verranno sommate per clienti donne e uomini con divisione per prodotto.
5. Ordino i valori risultanti delle due colonne dei clienti donne e uomini per **customer_id** con una funzione di ordinamento
6. Unisco nuovamente le due colonne in un'unica tabella con la funzione di merge join e sulla chiave **product_id**
7. Con una suddivisione condizionale seleziono solo i prodotti per i quali il numero delle **acquirenti donne è almeno una volta e mezza quello degli acquirenti uomini**
8. Scrivo su un file csv di destinazione il risultato, con una funzione destinazione file flat.