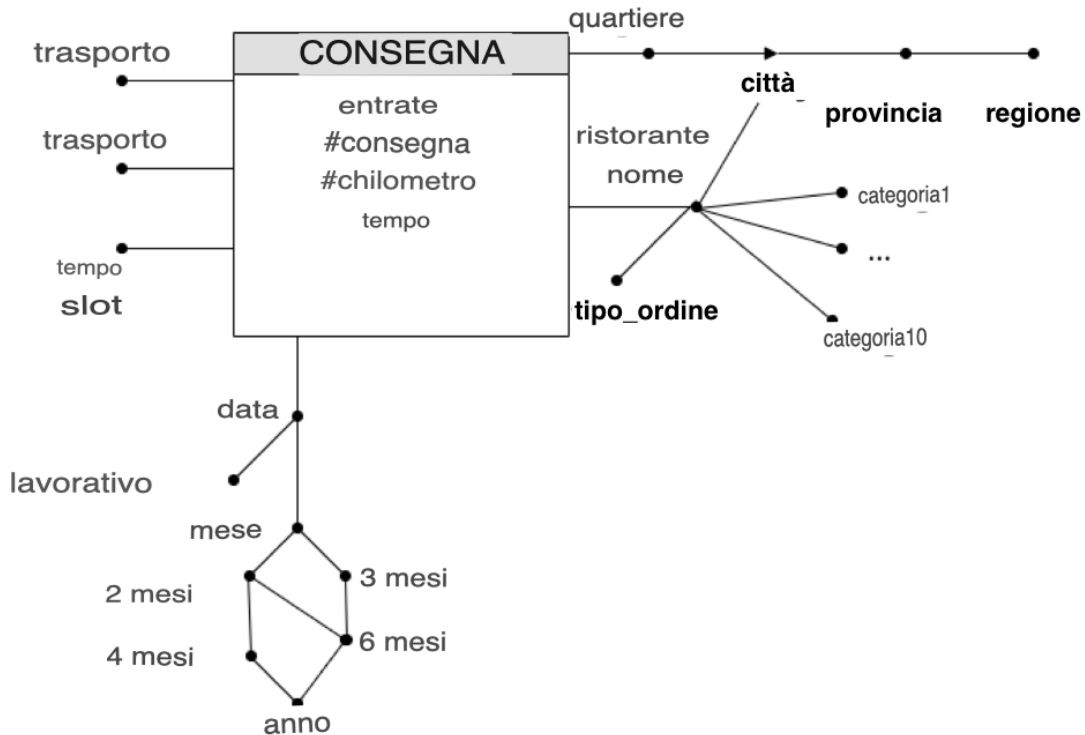


## Soluzione di data warehouse design



Le chiavi primarie sono sottolineate.

DELIVERIES (RIId, NIId, DIId, modalitàPagamento, trasporto, fasciaOraria, ricavo, #consegne, #chilometri, tempo\_consegna)

Ristorante (RIId, LIId, tipoOrdine, nome, categoria1, categoria2, ..., categoria10)

Località (LIId, città, provincia, regione)

Data (DIId, data, mese, 2-mesi, trimestre, 4-mesi, 6-mesi, anno, giornoLavorativo)

Quartiere (NIId, quartiere, LIId)

Soluzione Alternativa

DELIVERIES (RIId, LIId, DIId, modalitàPagamento, trasporto, fasciaOraria, ricavo, #consegne, #chilometri, tempo)

Ristorante (RIId, LIId, tipoOrdine, nome, categoria1, categoria2, ..., categoria10)

Località (LIId, città, provincia, regione, quartiere)

Data (DIId, data, mese, 2-mesi, trimestre, 4-mesi, 6-mesi, anno, giornoLavorativo)

## Queries

a	<pre>SELECT trasporto, trimestre,        SUM(numero_ore) / SUM(numero_consegne),        SUM(numero_consegne) / COUNT(DISTINCT timeslot, data),        RANK() OVER (ORDER BY SUM(numero_chilometri) / SUM(numero_ore) DESC) FROM Tempo T, Consegne C, Ristorante R WHERE c.id_ristorante = r.id_ristorante AND c.id_tempo = t.id_tempo       AND tipo_ordine = 'ordine_partner' GROUP BY trasporto, trimestre;</pre>
b	<pre>SELECT metodo_pagamento, città_destinazione, mese, trimestre,        SUM(SUM(rivvenuta)) OVER (PARTITION BY trimestre, metodo_pagamento, città_destinazione                                 ORDER BY mese ROWS UNBOUNDED PRECEDING),        SUM(rivvenuta) / SUM(numero_consegne),        100 * SUM(rivvenuta) / SUM(SUM(rivvenuta)) OVER (PARTITION BY mese, città_destinazione) FROM Tempo T, Consegne C, Ristorante R, Posizione L WHERE c.id_consegna = t.id_consegna AND c.id_ristorante = r.id_ristorante AND c.id_posizione = l.id_posizione AND r.pizza = true GROUP BY mese, trimestre, metodo_pagamento, città_destinazione;</pre>