



Data Base and Data Mining Group of Politecnico di Torino

Data Warehousing and Data Mining

Politecnico di Torino - School of Information Engineering
Master of Science in Computer Engineering

Exam 01–Feb–2008

Surname	
Name	
Student ID	

Oral exam dates:

- ☐ Monday, **February 11th**, 2007 – from 2 pm to 6 pm
- ☐ Tuesday, **February 12th**, 2007 – from 9 am to 13:30 pm

If you have a preference between the two dates, please indicate your choice ticking it off. Such preference, when possible, will be taken into account to prepare the oral session timetable.

Oral session timetables will be published on the course website by February 8th, 2007.

Design a data warehouse to analyze the business of an international parcels service with more than 200 branches all over the world, addressing the following issues.

Problem specifications

The branches manage dispatches of different products for many third party companies. The current information system is heterogeneous, thus each branch has its own data for its own business activities.

Each dispatch is handled by a single branch. The dispatch consists of one or more routes and every route has a departure and a destination place (city, province, region, and state).

The management needs to analyze the global flows of delivered goods to decide which branches to expand or reduce and to take strategic business decisions.

The analysis of the good flows is performed taking into account the income, the weight (expressed in kg), and the volume of the delivered goods. In particular, the management is interested in analyzing the income of each branch and in comparing it to the income of each district to which the branch belongs. A further analysis is performed on the profitability for different categories of goods and for different carrier types (air, rail, sea freight, etc).

To decide which branches to expand or reduce, the management needs to analyze the shipping routes in terms of income, volume, and weight of the delivered goods.

Eventually, the management needs to analyze the average income, the average weight, and the average volume of goods delivered in different years, semesters, 4-month periods, trimesters, 2-month periods, months, days of the month and days of the week.

The following are **some** of the frequent queries the management is interested in:

- a) Considering only Italian routes, for each category of goods and for each year, select the average daily income for each month and the total monthly income since the beginning of the year.
- b) Considering only air carriers, select the yearly average income per unit of volume for each destination province, and the percentage of such income compared to the yearly average income per unit of volume of the destination state.
- c) In 2006 for each route, in terms of departure and destination region, select the monthly average income per unit of weight (in kg) of the delivered goods, and the average daily income for each month of the goods delivered on that route.
- d) For each branch district and carrier type, select the total income for each month and the total volume of goods delivered in each month. Rank the results according to the total monthly volume (the highest is 1st).
- e) In 2005, for each route, in terms of departure and destination city, select the yearly average income per unit of weight of the delivered goods, and the average daily income for each considered route.
- f) In 2005 and 2006, considering only shipping by sea, select the half-year average income per unit of volume for each branch and for each departure region, and the half-year average income per unit of weight.
- g) For each departure city and for each branch district, select the 4-month period income and the total volume of delivered goods for each 4-month period.

Design

The data warehouse will store information of 2004, 2005, 2006 and 2007. The following cardinalities are known (suppose data is uniformly distributed):

- Good categories: ~20
 - Branches: ~200
 - Branch districts: ~50
 - Different carrier types: ~5
 - Cities: ~1000
 - Provinces: ~200
 - Regions: ~100
 - States: ~20
1. Design the data warehouse to address the described issues. In particular, the designed data warehouse must allow efficient execution of **all** the queries described in the specifications.
 2. Write the frequent queries (**a**), (**c**) and (**d**) of the “problem specifications” using the extended SQL language.
 3. Considering the designed data warehouse and its cardinalities, decide whether and which materialized views are convenient to improve response time of the frequent queries (consider **all** the frequent queries). Explain reasons for your choices.

```

/* #####*
* Sistemi per la gestione di basi di dati *
* Esame del 01-02-2008 ENG *
* Clemenza Carmelo - matricola: 147993 *
* #####*
*/

```

CARRIER_TYPE, GOOD_CATEGORY DIMENSIONI DEGENERI CON CARDINALITA RIDOTTA -> JUNK DIMENSION
IN ALTERNATIVA SI PUÒ FARE IL PUSH DOWN DI CARRIER_TYPE NELLA FACT **TABLE**, PERCHÈ DAI VALORI
PRESENTI NEL TESTO SI PUÒ SUPPORRE ESSERE UNA STRINGA DI LUNGHEZZA BREVE, ED AVERE UNA JUNK
DIMENSION CON SOLO LA DIMENSIONE GOOD_CATEGORY, CHE DAI VALORI PRESENTI NEL TESTO SI PUÒ
SUPPORRE ESSERE UNA STRINGA DI LUNGHEZZA MEDIO-ALTA.

DATA LA ELEVATA CARDINALITA DELLA TABELLA DEI FATTI E LA RIDOTTA CARDINALITA DELLE **2**
DIMENSIONI, SCEGLIAMO LA PRIMA SOLUZIONE.

```

DATE(COD_D, DATE, DAY_OF_THE_WEEK, DAY_OF_THE_MONTH, MONTH, TWO-M, THREE-M, FOUR-M, SEMESTER,
YEAR); -> CARD~=365*4=1.460
PLACES(COD_P, CITY, PROVINCE, REGION, STATE); -> CARD~=1.000
BRANCHES(COD_B, BRANCH, DISTRICT); -> CARD~=200
CARRIER_TYPE-GOOD_CATEGORY(COD_C, CARRIER_TYPE, GOOD_CATEGORY); -> CARD~=20*5=100
ROUTES(COD_D, COD_DEPARTURE, COD_DESTINATION, COD_B, COD_C, VOLUME, INCOME, WEIGHT, N_GOODS);
-> CARD~=1.460*(1000^2)*200*100~=3*10^13

```

NOTE: SAREBBE OPPORTUNO AGGIUNGERE INFORMAZIONE SUI 'DISPATCHES', NEL PROBLEMA NON RICHIESTA
QUINDI IGNORATA

--a)

```
SELECT GOOD_CATEGORY, YEAR, MONTH, SUM(INCOME)/COUNT(DISTINCT DATE) AS AVERAGE_DAILY_INCOME,
       SUM(SUM(INCOME)) OVER (PARTITION BY GOOD_CATEGORY, YEAR
                               ORDER BY MONTH
                               ROWS UNBOUNDED PRECEDING) AS
                               TOTAL_MONTHLY_INCOME_SINCE_THE_BEGINNING_OF_THE_YEAR
FROM ROUTES R, CARRIER_TYPE-GOOD_CATEGORY C, DATE D, PLACES FROM, PLACES TO
WHERE R.COD_C=C.COD_C AND R.COD_D=D.COD_D AND R.COD_DEPARTURE=FROM.COD_P AND R.
COD_DESTINATION=TO.COD_P AND
       FROM.STATE='ITALY' AND TO.STATE='ITALY'
GROUP BY GOOD_CATEGORY, YEAR, MONTH;
```

--b)

```
SELECT TO.PROVINCE, TO.STATE, YEAR, SUM(INCOME)/SUM(VOLUME) AS
YEARLY_AVERAGE_INCOME_PER_VOLUME,
       100*SUM(INCOME)/SUM(SUM(INCOME)) OVER (PARTITION BY TO,STATE, YEAR) AS
       INCOME_PROVINCE_ON_INCOME_STATE,
       SUM(SUM(INCOME))/SUM(SUM(VOLUME)) OVER (PARTITION BY TO,STATE, YEAR) AS
       STATE_YEARLY_AVERAGE_INCOME_PER_VOLUME,
FROM ROUTES R, CARRIER_TYPE-GOOD_CATEGORY C, DATE D, PLACES FROM, PLACES TO
WHERE R.COD_C=C.COD_C AND R.COD_D=D.COD_D AND R.COD_DEPARTURE=FROM.COD_P AND R.
COD_DESTINATION=TO.COD_P AND
       CARRIER_TYPE='AIR'
GROUP BY TO.PROVINCE, TO.STATE, YEAR;
```

--c)

```
SELECT DISTINCT FROM.REGION, TO.REGION, MONTH, SUM(INCOME)/SUM(WEIGHT) AS
MONTHLY_AVERAGE_INCOME_PER_WEIGHT,
       SUM(INCOME)/COUNT(DISTINCT DATE) AS AVERAGE_DAILY_INCOME
FROM ROUTES R, DATE D, PLACES FROM, PLACES TO
WHERE R.COD_D=D.COD_D AND R.COD_DEPARTURE=FROM.COD_P AND R.COD_DESTINATION=TO.COD_P AND D.
YEAR=2006
GROUP BY P_DEP.REGION, P_DEST.REGION, MONTH;
```

--d)

```
SELECT DISTRICT, CARRIER_TYPE, MONTH, SUM(INCOME) AS TOTAL_INCOME, SUM(VOLUME) AS TOTAL_VOLUME
       RANK() OVER (PARTITION BY DISTRICT, CARRIER_TYPE
                   ORDER BY SUM(VOLUME) DESC) AS POSIZIONE_MESE
FROM ROUTES R, BRANCHES B, CARRIER_TYPE-GOOD_CATEGORY C
WHERE R.COD_B=B.COD_B AND R.COD_C=C.COD_C
GROUP BY DISTRICT, CARRIER_TYPE, MONTH;
```

Sistemi informativi per la Business Intelligence

Esame del 21 aprile 2008 – Compito 1

Nome	
Cognome	
Matricola	

Progettare un data warehouse per la gestione delle problematiche illustrate nei punti seguenti relative alla gestione di conferenze scientifiche.

Descrizione del problema

Una fondazione di ricerca vuole analizzare le revisioni degli articoli nelle varie conferenze internazionali per valutare il lavoro dei ricercatori e inoltre vuole valutare gli incassi ottenuti dalle iscrizioni alle conferenze.

Le associazioni di ricerca scientifica (es. IEEE, ACM, ecc.) organizzano periodicamente delle conferenze in varie nazioni del mondo. I ricercatori possono inviare degli articoli alle conferenze. Ogni articolo tratta un determinato argomento (es. "Data warehouse aziendali"), che appartiene a un preciso ambito di ricerca (es. "Informatica"). Lo stesso articolo può essere inviato a più conferenze ed è valutato da un insieme di revisori. Il compito dei revisori è quello di analizzare i diversi aspetti dell'articolo (es. l'originalità, la pertinenza, ecc.) e assegnare un voto a ogni aspetto.

I ricercatori che partecipano alle conferenze pagano una quota di iscrizione a seconda dei servizi ai quali si iscrivono (le sessioni della conferenza, i laboratori scientifici, la cena sociale, ecc.). Alcuni poi possono usufruire di sconti speciali a seconda della loro età, o della loro professione o della modalità di pagamento.

Gli analisti della fondazione di ricerca sono interessati ad analizzare il voto assegnato dai revisori agli articoli inviati alle conferenze in funzione:

- della conferenza alla quale l'articolo è stato inviato
- del mese e dell'anno in cui si è tenuta la conferenza
- dell'associazione organizzatrice della conferenza
- della nazione e del continente in cui si è tenuta la conferenza
- del tipo di valutazione (originalità, pertinenza, ecc.)
- del revisore
- dell'articolo inviato
- dell'argomento e dell'ambito di ricerca dell'articolo.

Inoltre, per ogni partecipante, la fondazione di ricerca vuole analizzare la quota totale e lo sconto totale effettuato in funzione:

- della conferenza
- dell'associazione organizzatrice della conferenza
- del mese e dell'anno in cui si è tenuta la conferenza
- del partecipante, del suo sesso e del suo anno di nascita
- della professione del partecipante
- della modalità di pagamento (bancomat, contanti, ecc.)
- dei servizi ai quali il partecipante si è iscritto (sessioni, laboratori, cena sociale, ecc.).

Il data warehouse realizzato deve contenere le informazioni relative agli anni 1998-2007. Al fine di una corretta realizzazione del data warehouse sono state fornite le seguenti informazioni:

- o Numero di conferenze ~ 10 000
- o Numero di nazioni in cui sono state svolte conferenze ~ 100
- o Numero di associazioni organizzatrici ~ 100
- o Numero di revisori ~ 1000
- o Numero di valutazioni ~ 10
- o Numero di articoli ~ 100 000
- o Numero di argomenti di ricerca ~ 1000
- o Numero di ambiti di ricerca ~ 20
- o Numero di partecipanti ~ 10 000
- o Numero di professioni dei partecipanti ~ 50

Sono di seguito riportate **alcune** delle interrogazioni frequenti di interesse per la fondazione di ricerca:

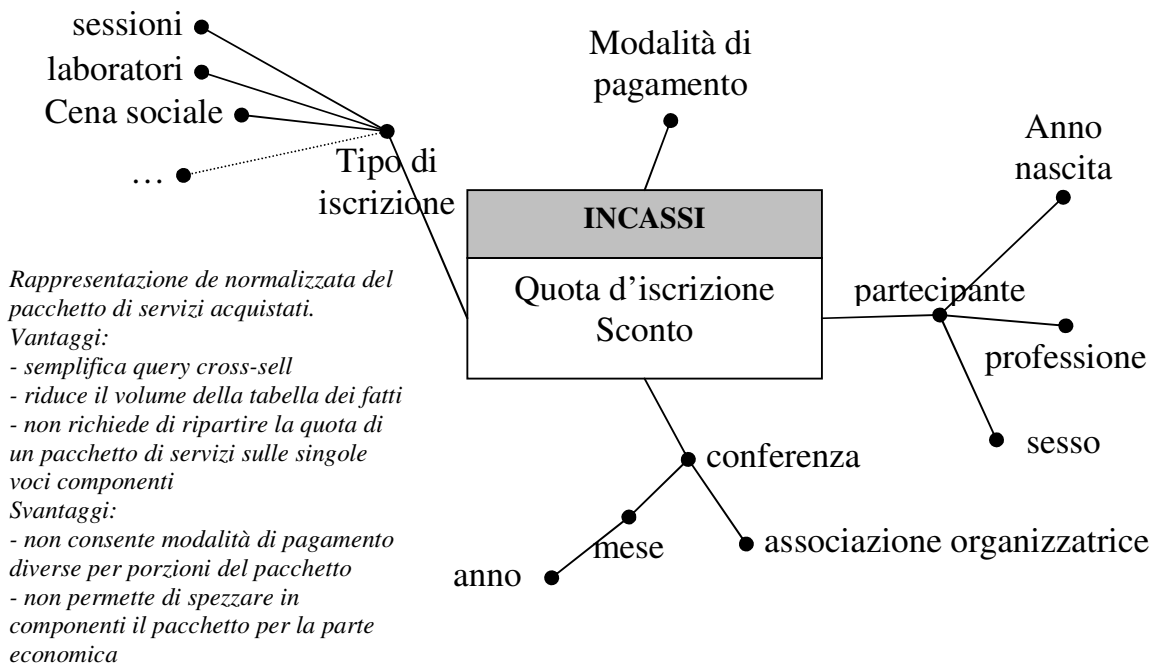
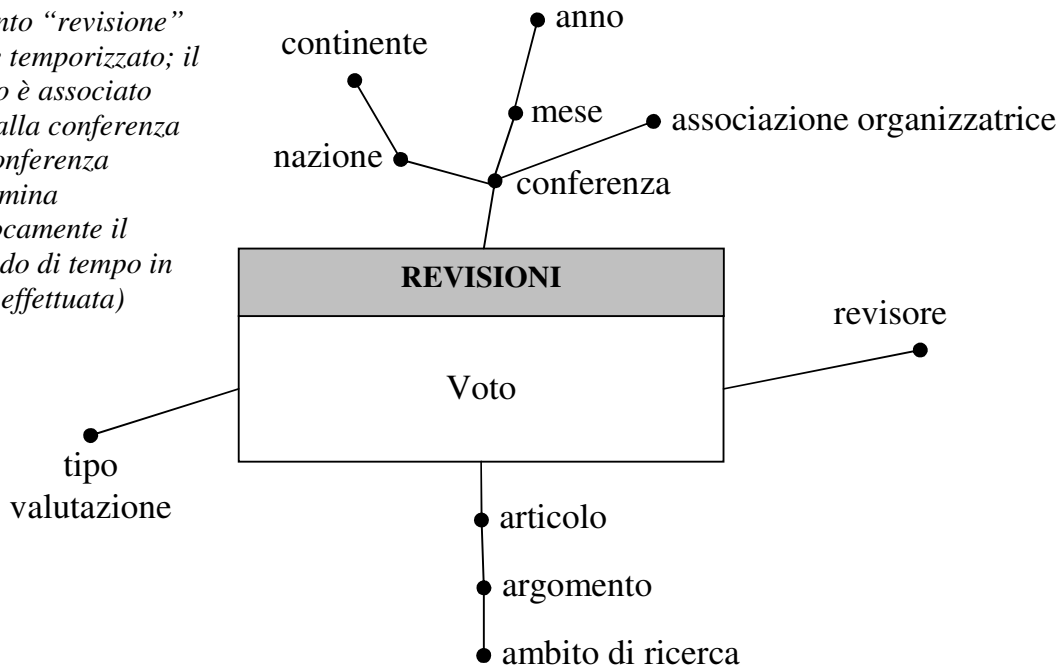
- a) Considerando solo l'anno 2007, per le conferenze a cui sono stati inviati almeno 500 articoli nell'ambito di ricerca "Informatica", visualizzare la conferenza, l'argomento e il numero totale di articoli inviati sull'argomento.
- b) Considerando solo gli articoli che trattano l'argomento "Data warehouse", per ogni conferenza e per ogni tipo di valutazione, visualizzare la differenza tra il voto massimo degli articoli della conferenza e il voto massimo degli articoli di tutte le conferenze tenute nello stesso anno.
- c) Considerando le revisioni di conferenze tenute in Europa, per tutti gli articoli con voto medio di originalità superiore a 5, visualizzare il rapporto tra il voto medio di originalità di ogni articolo e il voto medio di originalità degli articoli dello stesso argomento. Ordinare gli articoli in base ai valori del precedente rapporto.
- d) Ordinare, assegnando un rank, le conferenze tenute in Europa negli anni 2004 e 2005 per voto medio di originalità decrescente, considerando solo gli articoli dell'ambito di ricerca "Informatica".
- e) Per l'anno 2008, visualizzare l'incasso mensile e l'incasso mensile cumulativo dall'inizio dell'anno, separatamente per ogni modalità di pagamento e professione dei partecipanti.
- f) Per ogni associazione organizzatrice e per ogni anno, visualizzare la percentuale media di sconto applicata per ciascuna modalità di pagamento.
- g) Per ogni conferenza organizzata dall'associazione IEEE, visualizzare la percentuale di incassi derivante dalle iscrizioni in base alla professione dei partecipanti (es. 32% degli incassi dovuti a iscrizioni di partecipanti "Ingegneri" e 10% degli incassi derivanti da iscrizioni di partecipanti "Analisti"). Considerare nell'analisi solo i partecipanti che si sono iscritti sia alle sessioni, sia alla cena sociale.
- h) Considerando solo le conferenze tenute in Europa negli anni 2004 e 2005, e gli articoli che trattano l'argomento "Data warehouse", visualizzare il voto medio assegnato ad ogni articolo per ciascun tipo di valutazione. Ordinare i risultati per voto medio decrescente.

Progettazione

- 1. (12 PUNTI) Progettare il data warehouse necessario per analizzare le pubblicazioni in modo da soddisfare le richieste descritte nelle specifiche del problema. Il data warehouse progettato deve inoltre permettere di rispondere in modo efficiente a **tutte** le interrogazioni frequenti proposte nelle specifiche del problema.
- 2. (15 PUNTI) Esprimere le interrogazioni frequenti (**a**), (**e**), (**f**) delle specifiche del problema utilizzando il linguaggio SQL esteso.
- 3. (2 PUNTI) Considerando le caratteristiche del data warehouse realizzato e la cardinalità dei dati memorizzati nel data warehouse, decidere se e quali viste materializzate potrebbe essere utile definire al fine di ottimizzare i tempi di risposta delle interrogazioni proposte nelle specifiche del problema (considerare **tutte** le interrogazioni proposte e non solo quelle risolte in SQL al punto 2). Motivare le scelte fatte.
- 4. (1 PUNTO) Decidere come gestire la dinamicità (variazione) dei dati all'interno delle dimensioni.

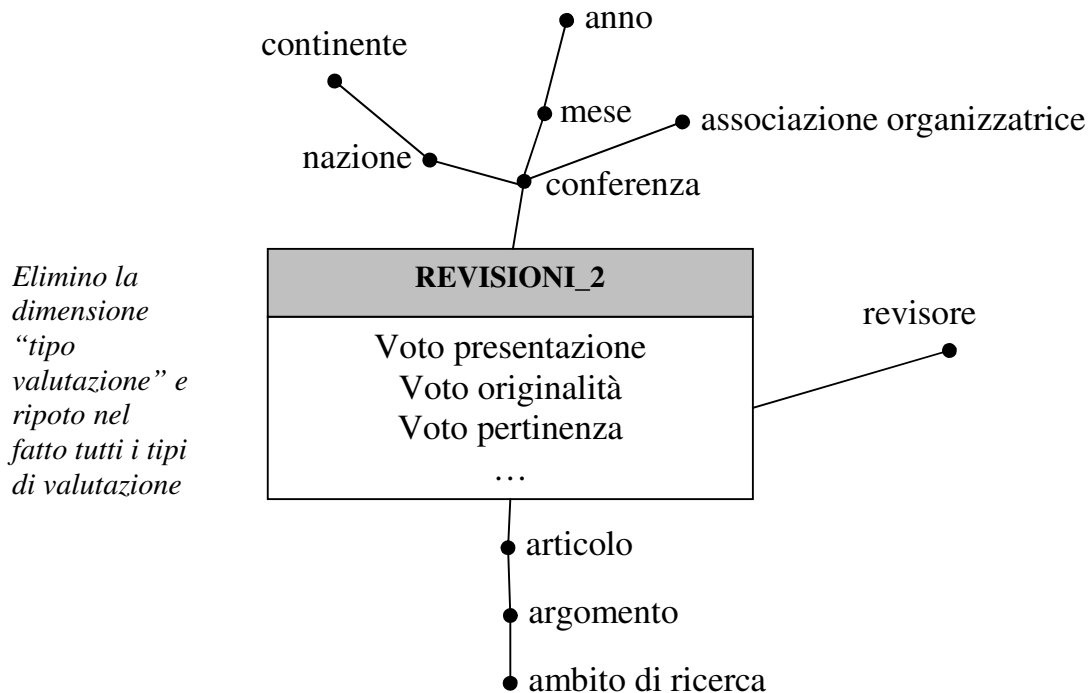
Progettazione concettuale

*l'evento "revisione"
non è temporizzato; il
tempo è associato
solo alla conferenza
(la conferenza
determina
univocamente il
periodo di tempo in
cui è effettuata)*



Variente fatto revisioni

Elimino la dimensione “tipo valutazione” e riporto nel fatto tutti i tipi di valutazione



Vantaggi della seconda soluzione

- Minor volume della tabella dei fatti (una sola riga per articolo al posto che N righe una per ogni tipo di valutazione)

Svantaggi della seconda soluzione

- Meno flessibile
 - o Per i nuovi eventuali tipi di valutazione
 - o Per le conferenze con parametri di valutazione diversi

Rappresentazione normalizzata del pacchetto di servizi acquistati.

Vantaggi:

- semplifica query cross-sell
- riduce il volume della tabella dei fatti
- non richiede di ripartire la quota di un pacchetto di servizi sulle singole voci componenti

Svantaggi:

- non consente modalità di pagamento diverse per porzioni del pacchetto
- non permette di spezzare in componenti il pacchetto per la parte economica

Fatti:

Revisioni (di articoli a conferenza)

CONFERENZA1 (CodC1, mese, anno, organizzazione, nazione, continente)

REVISORE (CodR, nome_revisore)

TIPO_VALUTAZIONE (CodT, tipo_valutazione)

ARTICOLO (CodA, titolo_articolo, argomento, ambito_ricerca)

REVISIONE(CodC1, CodR, CodT, CodA, voto)

Cardinalità:

1000 revisori x 10 valutazioni x 10 000 conferenze x 100 000 articoli

(in realtà gli eventi sono più sparsi)

Iscrizioni

CONFERENZA2 (CodC2, mese, anno, organizzazione)

PARTECIPANTE (CodP, nome, età, professione, sesso)

MODALITA_PAGAMENTO(CodM, modo_pagamento)

TIPO_ISCRIZIONE (CodT, sessione, laboratorio, cena,...)

INCASSI(CodP, CodC2, CodM, CodT, quota, Sconto)

CodM può fare parte o no della chiave prima di INCASSO (se no, ho una sola modalità di pagamento)

Cardinalità:

10 000 conferenze x 10 000 partecipanti x 10 modalità di pagamento x 100 tipi di iscrizione

(in realtà gli eventi sono più sparsi)

Uniformazione delle dimensioni

La dimensione CONFERENZA viene condivisa da entrambi i fatti

Dim:

CONFERENZA (CodC, mese, anno, organizzazione, nazione, continente)

REVISORE (CodR, nome_revisore)

TIPO_VALUTAZIONE (CodT, tipo_valutazione)

ARTICOLO (CodA, titolo_articolo, argomento, ambito_ricerca)

PARTECIPANTE (CodP, nome, età, professione, sesso)

MODALITA_PAGAMENTO(CodM, modo_pagamento)

TIPO_ISCRIZIONE (CodT, sessione, laboratorio, cena,...)

Fact:

REVISIONE(CodC, CodR, CodT, CodA, voto)

INCASSI(CodP, CodC, CodM, CodT, quota, Sconto)

Query (compito 1)

Considerando solo l'anno 2007, per le conferenze a cui sono stati inviati almeno 500 articoli nell'ambito di ricerca "Informatica", visualizzare la conferenza, l'argomento e il numero totale di articoli inviati sull'argomento.

```
SELECT C.conferenza, A.argomento, count(distinct A.articolo)
FROM REVISIONE R, CONFERENZA C, ARTICOLO A
WHERE R.CodC=C.CodC AND R.CodA=A.CodA
AND C.ID IN
(
SELECT C.ID
FROM REVISIONI R, CONFERENZA C, ARTICOLO A
WHERE R.CodC=C.CodC AND R.CodA=A.CodA
AND C.ANNO=2007
AND A.AMBITO="informatica"
GROUP BY C.ID
HAVING count(distinct A.articolo)>=500
)
GROUP BY C.conferenza, A.argomento
```

La conferenza indica univocamente un anno, quindi nella query esterna non è necessario aggiungere la condizione sul tempo

Per l'anno 2007, visualizzare l'incasso mensile e l'incasso mensile cumulativo dall'inizio dell'anno, separatamente per ogni modalità di pagamento e professione dei partecipanti.

```
SELECT mese, professione, modalita_pag, SUM(quota)-SUM(sconto),
       SUM(SUM(quota)-SUM(sconto)) OVER
       (PARTITION BY professione,modalita_pag
        ORDER BY Mese ROWS UNBOUNDED PRECEDING)
FROM INCASSI I, CONFERENZA C, MODALITA_PAGAMENTO M, PARTECIPANTE P
WHERE I.CodC=C.CodC AND I.CodM=M.CodM AND I.CodP=P.CodP AND
AND anno=2007
GROUP BY mese, professione, modalità_pag
```

Per ogni associazione organizzatrice e per ogni anno, visualizzare la percentuale media di sconto applicata per ciascuna modalità di pagamento.

```
SELECT associazione, anno, modalitàP, SUM( sconto ) / SUM( quota ) * 100
FROM INCASSI I, CONFERENZA C, MODALITA_PAGAMENTO M
WHERE I.CodC=C.CodC AND I.CodM=M.CodM AND
GROUP BY associazione, anno, modalitàP
```

Media sull'individuo!!

```
SELECT associazione, anno, modalitàP, AVG( sconto/quota ) * 100
FROM INCASSI I, CONFERENZA C, MODALITA_PAGAMENTO M
WHERE I.CodC=C.CodC AND I.CodM=M.CodM AND
GROUP BY associazione, anno, modalitàP
```

Viste

Numero modalità di pagamento = 10

Numero di servizi= 10

Cardinalità della tabella TIPO_ISCRIZIONE = $2^{10} = 1024$

Cardinalità fatto REVISIONI: $100k \text{ (art)} * 10k \text{ (conf)} * 1000 \text{ (rev)} * 10 \text{ (val)} = 10000 \text{ Miliardi}$

Cardinalità fatto INCASSI: $1024 \text{ (servizi)} * 10k \text{ (part)} * 10 \text{ (mod.pag.)} * 10k \text{ (conf)} = 1024 \text{ Miliardi}$

Query	Group By	Predicati
A	CodC, Ambito, Argomento, (articolo)	Anno
B	CodC, TipoValutazione, Anno	Argomento
C	Articolo, Argomento	Continente, tipo Valutazione
D	CodC	Continente, ambito, tipo valutazione, ambito
E	Mese, mod_pagamento, professione	Anno
F	Assoc_organiz, anno mod_pagamento	
G	CodC, professione	Assoc_organiz, tipo iscrizione (flag)
H	Articolo, tipo_valutazione	Continente, anno, argomento

Vista 1

Conferenza X professione X ModPagamento = $10k * 50 * 10 = 5 \times 10^6$

Risponde a query E-F

Sistemi informativi per la Business Intelligence

Esame del 9 giugno 2008

Nome	
Cognome	
Matricola	

Descrizione del problema

Il Politecnico di Torino richiede un'analisi sull'occupazione delle aule nelle varie sedi, al fine di poter aiutare l'Ufficio Logistica nella gestione degli spazi.

Il politecnico è interessato ad analizzare quali sono le aule "meno utilizzate" (e se questo deriva dal fatto che manchino alcune attrezzature fondamentali per fare lezione) e quelle "peggio utilizzate" (ossia di cui vengono utilizzati molti meno posti della capienza totale). E' richiesta un'analisi per le diverse sedi del politecnico e a seconda del periodo dell'anno. L'anno accademico (A.A.) è diviso in 2 periodi didattici (PD) e inizia e finisce solitamente in giorni diversi ogni anno (tipicamente dalla prima metà di settembre alla seconda metà di maggio). Le aule sono caratterizzate da una sede (es. sede centrale c.so Duca, Castello del Valentino, ...) e da un tipo (aula normale, laboratorio, aula da conferenza,...).

Si è quindi interessati ad analizzare il numero di ore in cui l'aula è occupata in valore assoluto e in valore percentuale rispetto al numero di ore in cui l'aula è disponibile, la percentuale di posti occupati rispetto a quelli disponibili in funzione:

- dell'anno accademico e del periodo didattico (es. 1°PD 2006/7, 2°PD 2006/7,...)
- del periodo didattico dell'anno (1°PD, 2°PD,...)
- della data, della settimana, del giorno della settimana, del mese e dell'anno solare
- dell'aula e della sua sede
- della città, provincia e regione della sede
- del tipo di aula e delle attrezzature presenti o assenti nell'aula (video proiettore, lavagna luminosa, microfono,...)

Inoltre si è interessati a calcolare il numero di ore in cui l'aula è occupata in funzione:

- del nome del corso (che usufruisce dell'aula) e della Facoltà di cui fa parte il corso
- dell'aula e del tipo di aula
- dell'anno accademico
- del tipo di attività che si svolge nell'aula (lezione, seminario, esame,...)

Il data warehouse realizzato deve contenere le informazioni relative agli anni accademici 2000/01-2007/08. Al fine di una corretta realizzazione del data warehouse sono state fornite le seguenti informazioni:

- o Numero di Aule ~ 1000
- o Numero di tipi di Aule ~ 5
- o Numero di corsi tenuti al Politecnico di Torino ~ 5000
- o Numero di Facoltà ~ 6
- o Numero di attività ~ 5

Sono di seguito riportate **alcune** delle interrogazioni frequenti di interesse:

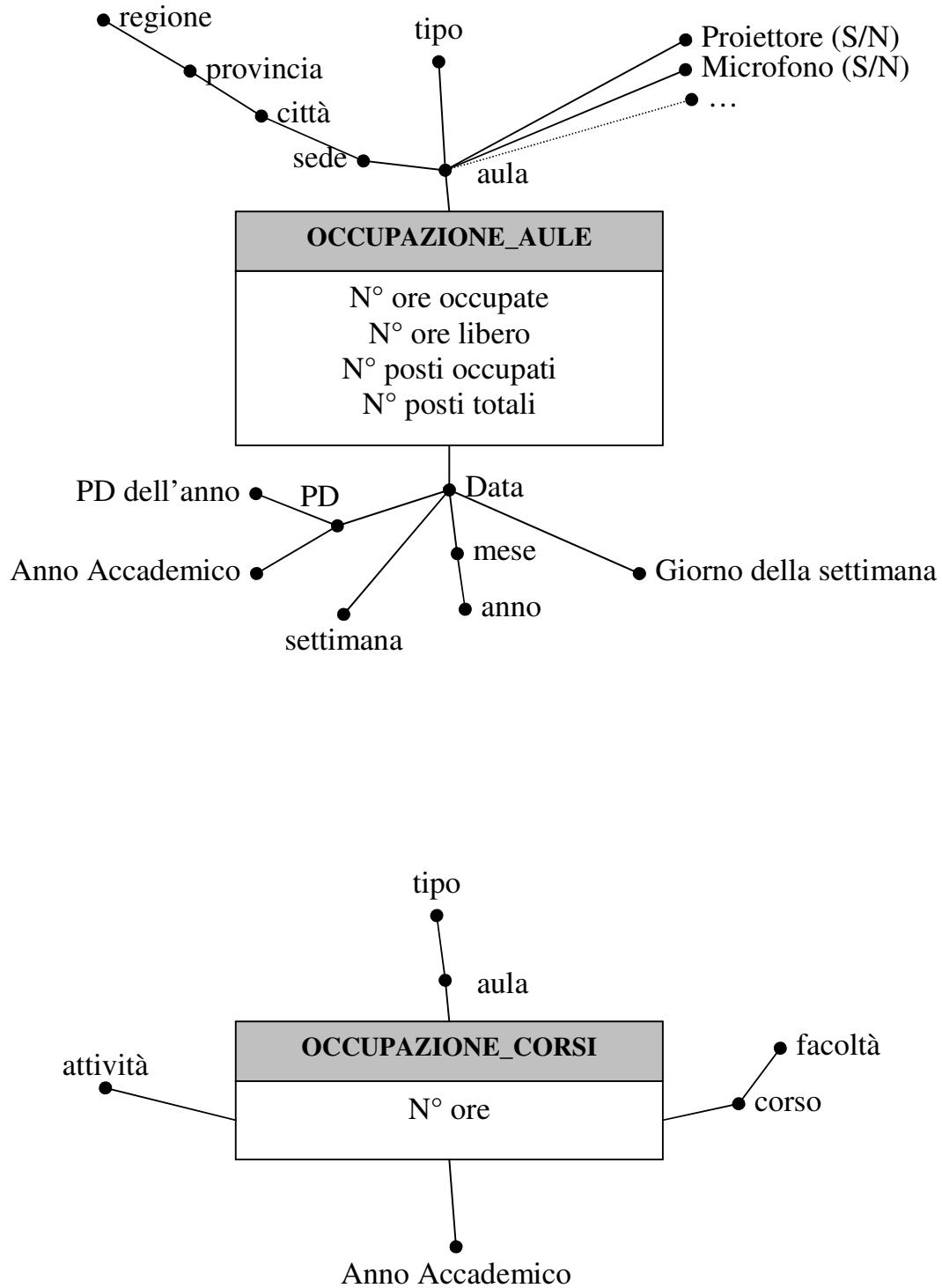
- a) Per ogni periodo didattico del A.A. 2006/07, trovare per ogni città il numero di ore totali in cui le aule sono state occupate
- b) Nel A.A. 2005/06 trovare per ogni regione e periodo didattico il numero di ore totali in cui le aule sono state libere
- c) Trovare il corso che nel a.a. 2005/06 ha passato più ore in una stessa aula.
- d) Per ogni anno accademico e per ogni corso trovare il numero di ore in cui sono state occupate aule di tipo "laboratorio"
- e) Per ogni corso trovare il numero di ore di lezione tenute nel a.a. 2006/07
- f) Relativamente all'anno accademico 2005/06 e considerando solo le aule che sono senza proiettore e che sono rimaste libere per più del 50% delle ore disponibili, calcolare per ogni aula la percentuale annuale di posti occupati
- g) Per i Corsi che nel A.A.2005/2006 hanno fatto più di 20 ore in laboratorio, trovare il numero di ore in laboratorio effettuate nel A.A.2006/2007

- h) Visualizzare la classifica delle Facoltà (nome facoltà e rank associato) che nell'A.A. 2005/2006 hanno fatto più ore di seminari (tipo di attività) in aule di tipo "conferenza". Ordinare le Facoltà da quella che a fatto più ore a quella che ne ha fatte di meno

Progettazione

1. (12 PUNTI) Progettare il data warehouse necessario per analizzare le pubblicazioni in modo da soddisfare le richieste descritte nelle specifiche del problema. Il data warehouse progettato deve inoltre permettere di rispondere in modo efficiente a **tutte** le interrogazioni frequenti proposte nelle specifiche del problema.
2. (15 PUNTI) Esprimere le interrogazioni frequenti **(f)**, **(g)**, **(h)** delle specifiche del problema utilizzando il linguaggio SQL o il linguaggio SQL esteso.
3. (2 PUNTI) Considerando le caratteristiche del data warehouse realizzato e la cardinalità dei dati memorizzati nel data warehouse, decidere se e quali viste materializzate potrebbe essere utile definire al fine di ottimizzare i tempi di risposta delle interrogazioni proposte nelle specifiche del problema (considerare **tutte** le interrogazioni proposte e non solo quelle risolte in SQL al punto 2). Motivare le scelte fatte.
4. (1 PUNTO) Decidere come gestire la dinamicità (variazione) dei dati all'interno delle dimensioni.

Progettazione concettuale



Fatti:

Occupazione Aule

AULA (CodA, aula, tipo, sede ,città, provincia, regione, proiettore, microfono, ...)

DATA (CodD, data, pd, pd_anno, a_a, settimana, mese, anno, giorno_settimana)

OCCUPAZIONE_AULE(CodA, CodD, n_ore_occupate, n_ore_libere, n_posti_occupati, n_posti_totali)

Occupazione Corsi

AULA2 (CodA2, aula, tipo)

CORSO (CodC, corso, facoltà)

A_A (CodAA, Anno_accademico)

ATTIVITA (CodAT, attività)

OCCUPAZIONE_CORSI(CodA2, CodC, CodAA, CodAT, n_ore)

Uniformazione delle dimensioni

Dim:

AULA (CodA, aula, tipo, sede ,città, provincia, regione, proiettore, microfono, ...)

DATA (CodD, data, pd, pd_anno, a_a, settimana, mese, anno, giorno_settimana)

CORSO (CodC, corso, facoltà)

A_A (CodAA, Anno_accademico)

ATTIVITA (CodAT, attività)

Fact:

OCCUPAZIONE_AULE(CodA, CodD, n_ore_occupate, n_ore_libere, n_posti_occupati, n_posti_totali)

OCCUPAZIONE_CORSI(CodA, CodC, CodAA, CodAT, n_ore)

Query

Relativamente all'anno accademico 2005/06 e considerando solo le aule che sono senza proiettore e che sono rimaste libere per più del 50% delle ore disponibili, calcolare per ogni aula la percentuale annuale di posti occupati

```
SELECT AULA, SUM(n_posti_occupati)/SUM(n_posti_totali)*100
FROM OCCUPAZIONE_AULE, AULE, DATA
WHERE OCCUPAZIONE_AULE.CodA=AULE.CodA AND
OCCUPAZIONE_AULE.CodD=DATA.CodD
AND proiettore="N"
AND anno_accademico="2005/2006"
GROUP BY AULA
HAVING SUM(n_ore_occupate)/(SUM(n_ore_libere)+SUM(n_ore_occupate))<0,5
```

Per i Corsi che nel A.A.2005/2006 hanno fatto più di 20 ore in laboratorio, trovare il numero di ore in laboratorio effettuate nel A.A.2006/2007

```
SELECT corso SUM(n_ore)
FROM OCCUPAZIONE_CORSI, CORSO, AULE, ANNO_ACCADEMICO
WHERE OCCUPAZIONE_CORSI.CodC=CORSO.CodC AND
OCCUPAZIONE_CORSI.CodA=AULE.CodA AND
OCCUPAZIONE_CORSI.CodAA=ANNO_ACCADEMICO.CodAA
AND tipo="laboratorio"
AND anno_accademico="2006/2007"
AND ID_corso IN
(
    SELECT ID_corso
    FROM OCCUPAZIONE_CORSI, AULE, ANNO_ACCADEMICO
    WHERE ...
    AND tipo="laboratorio"
    AND anno_accademico="2005/2006"
    GROUP BY ID_corso
    HAVING SUM(n_ore)>20
)
GROUP BY corso
```

Visualizzare la classifica delle Facoltà (nome facoltà e rank associato) che nell'anno 2005/2006 hanno fatto più ore di seminari in aule di tipo "conferenza". Ordinare le Facoltà da quella che a fatto più ore a quella che ne ha fatte di meno

```
SELECT facolta, RANK() OVER (ORDER BY SUM(n_ore) DESC) as classifica
FROM OCCUPAZIONE_CORSI, CORSO, ATTIVITA, AULA
WHERE OCCUPAZIONE_CORSI.CodC=CORSO.CodC AND
OCCUPAZIONE_CORSI.CodA=AULE.CodA AND
OCCUPAZIONE_CORSI.CodAA=ANNO_ACCADEMICO.CodAA AND
OCCUPAZIONE_CORSI.CodAT=ATTIVITA.CodAT
AND anno_accademico="2005/2006"
AND attività="seminario"
AND tipo="conferenza"
GROUP BY facolta
ORDER BY SUM(n_ore) DESC
```


Viste

Date = 9 (mesi circa) * 31 (giorni) * 9 (anni) = ~ 2500

Cardinalità fatto OCCUPAZIONE_AULE: 2200*1000 = ~ 2 Milioni

Cardinalità fatto OCCUPAZIONE_CORSI: 1000*5000*5*8 = ~ 200 Milioni

Query	Group By	Predicati
A	Pd, città	a.a.
B	Regione, pd	a.a.
C	Aula, corso	a.a.
D	a.a. , corso	Tipo_aula
E	corso	Attività, a.a.
F	aula	Proiettore, a.a.
G	corso	Tipo_aula, a.a.
H	facolta	a.a., attività, tipo_aula

```
/* #####
* ## Sistemi per la gestione di basi di dati      ##
* ## Trigger esame del 17-09-2010                ##
* ## Clemenza Carmelo - matricola: 147993         ##
* #####
*/

--TRIGGER 1: Assegnazione di una piazzola
CREATE OR REPLACE TRIGGER RICHIESTA_PIAZZOLA
BEFORE INSERT
ON RICHIESTA_PIAZZOLA
FOR EACH ROW
DECLARE
    SCADENZA DATE;
    MIN_COD_PIAZZOLA NUMBER;
BEGIN
    --Verifico se la quato associativa del cliente e' ancora attiva
    SELECT DataScadenzaQuotaAssociativa INTO SCADENZA
    FROM CLIENTI
    WHERE CodCliente=:NEW.CodCliente;

    IF (DATE(:NEW.TimeStampIngresso)>SCADENZA) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Errore: Quota associativa cliente scaduta');
    END IF;

    --Verifico se esiste una piazzola disponibile
    SELECT MIN(CodP) INTO MIN_COD_PIAZZOLA
    FROM PIAZZOLA
    WHERE StatoPiazzola='LIBERO';

    IF (MIN_COD_PIAZZOLA IS NULL) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Errore: Nessuna piazzola disponibile');
    END IF;

    --Assegno la piazzola al cliente che ne ha fatto richiesta
    UPDATE PIAZZOLA
    SET StatoPiazzola='OCCUPATO', TargaVettura=:NEW.TargaVettura, CodCliente=:NEW.CodCliente,
        TimeStampIngresso=:NEW.TimeStampIngresso
    WHERE CodP=MIN_COD_PIAZZOLA;
END;
/

--#####
```

```
--TRIGGER 2: Rilascio di una piazzola precedentemente assegnata
CREATE OR REPLACE TRIGGER RILASCIO_PIAZZOLA
AFTER INSERT
ON RILASCIO_PIAZZOLA
FOR EACH ROW
DECLARE
    INGRESSO TIMESTAMP;
    MAX_COD_N NUMBER;
BEGIN
    --Memorizzo l'istante di ingresso nel parcheggio
    SELECT TimeStampIngresso INTO INGRESSO
    FROM PIAZZOLA
    WHERE TargaVettura=:NEW.TargaVettura;

    IF(INGRESSO IS NULL) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Errore: Il cliente non era entrato nel parcheggio');
    END IF;

    --Aggiorno l'informazione della piazzola liberata
    UPDATE PIAZZOLA
    SET StatoPiazzola='LIBERO', TargaVettura=NULL, CodCliente=NULL, TimeStampIngresso=NULL
    WHERE TargaVettura=:NEW.TargaVettura;

    --Notifico l'intervallo di tempo di parcheggio al cliente
    SELECT MAX(CodN) INTO MAX_COD_N
    FROM NOTIFICA;

    IF(MAX_COD_N IS NULL) THEN
        MAX_COD_N:=1;
    ELSE
        MAX_COD_N:=MAX_COD_N+1;
    END IF;

    INSERT INTO NOTIFICA(CodN, CodCliente, TargaVettura, IntervalloTempoParcheggio)
    VALUES(MAX_COD_N, :NEW.TargaVettura, :NEW.TimeStampUscita-INGRESSO);
END;
/

--#####

--TRIGGER 3: Gestione della quota associativa al parcheggio
CREATE OR REPLACE TRIGGER QUOTA_ASSOCIATIVA
BEFORE INSERT OR UPDATE OF QuotaAssocitiva
ON CLIENTI
FOR EACH ROW
WHEN (NEW.QuotaAssocitiva<300)
BEGIN
    --Correggo il valore errato di quota associativa
    :NEW.QuotaAssocitiva:=300;
END;
/

--#####
```

```
--TRIGGER 4: Calcolo statistiche sull'utilizzo del parcheggio
CREATE OR REPLACE TRIGGER CALCOLO_STATISTICHE
BEFORE INSERT
ON RICHIESTA_STATISTICHE
FOR EACH ROW
DECLARE
    N_PARCHEGGI NUMBER;
    N_VETTURE_DIVERSE NUMBER;
    DURATA_TOTALE_PARCHEGGI TIMESTAMP;
BEGIN
    --Calcolo le statistiche dalla tabella NOTIFICA
    SELECT COUNT(*), COUNT(DISTINCT TargaVettura), SUM(IntervalloTempoParcheggio) INTO
    N_PARCHEGGI, N_VETTURE_DIVERSE, DURATA_TOTALE_PARCHEGGI
    FROM NOTIFICA
    WHERE CodCliente=:NEW.CodCliente;

    --Inserisco le statistiche nella tabella NOTIFICA_STATISTICHE
    INSERT INTO NOTIFICA_STATISTICHE(CodStatistica, CodCliente, NumeroParcheggi,
    NumeroVettureDiverse, DurataMediaParcheggio)
    VALUES (:NEW.CodStatistica, :NEW.CodCliente, N_PARCHEGGI, N_VETTURE_DIVERSE,
    DURATA_TOTALE_PARCHEGGI/N_PARCHEGGI);
END;
/
```

2 febbraio 2006

Appello di

Analisi di Basi di dati - Progettazione

Progettare un data warehouse per la gestione delle problematiche illustrate nei punti seguenti relative ad una catena di alberghi.

1. Descrizione del problema

Una grande catena di alberghi gestisce circa 500 alberghi, di diverse categorie, sparsi in tutto il mondo. Nella base di dati di ogni albergo sono memorizzate tutte le informazioni giornaliere relative ai clienti che soggiornano nell'albergo e in quali camere. Per ogni albergo sono note, sempre a livello giornaliero, non solo le informazioni su quali camere sono occupate, ma anche quali camere sono disponibili (libere) e quali non sono agibili perché si stanno facendo delle riparazioni alla camera o all'arredamento.

La dirigenza della catena di alberghi vuole disporre di un data warehouse che permetta di effettuare analisi di mercato relativamente agli incassi ottenuti dai vari alberghi in funzione:

- della zona geografica in cui si trova l'albergo (stato, regione, città)
- della categoria dell'albergo (5 stelle, 4 stelle, ..)
- delle caratteristiche della camera (numero di letti, dotata di televisore, dotata di vasca idromassaggio, ..)
- della data, del giorno della settimana, del mese e dell'anno

La dirigenza della catena di alberghi vuole inoltre sapere a livello giornaliero per ogni albergo:

- la frazione di camere occupate
- la frazione di camere libere
- la frazione di camere non agibili a causa di lavori di riparazione

Anche relativamente alla frazione di camere nei diversi stati (occupate, libere, non agibili) le informazioni devono essere disponibili in funzione:

- della zona geografica in cui si trova l'albergo
- della categoria dell'albergo (5 stelle, 4 stelle, ..)
- delle caratteristiche della camera (numero di letti, dotata di televisore, dotata di vasca idromassaggio, ..)
- della data, del giorno della settimana, del mese e dell'anno

La dirigenza dell'albergo vuole disporre delle informazioni appena descritte sia a livello giornaliero, sia a livello mensile, sia a livello annuale. Si vuole inoltre poter capire come variano gli incassi in funzione del fatto che il giorno sia festivo oppure no.

Sono di seguito riportate alcune delle interrogazioni frequenti di interesse per la dirigenza della catena di alberghi:

- a) Relativamente all'anno 2005, selezionare per ogni coppia (stato, mese) la frazione mensile di camere occupate sulla totalità di camere, la frazione mensile di camere libere sulla totalità di camere e la frazione mensile di camere non agibili sulla totalità di camere.
- b) Relativamente all'anno 2005, selezionare per ogni stato la frazione di camere occupate sulla totalità di camere in ogni stato nel corso dell'anno 2005. Associare ad ogni stato un attributo di rank associato alla frazione di camere occupate sulla totalità di camere nello stato nel corso dell'anno 2005. L'attributo di rank deve assumere il valore 1 per lo stato con la maggiore frazione di camere occupate sulla totalità di camere nello stato nel corso dell'anno 2005.
- c) Relativamente all'anno 2005, selezionare per ogni coppia (stato, mese) l'incasso mensile calcolato considerando solo gli alberghi a 4 stelle e l'incasso cumulativo da inizio anno (sempre considerando solo gli alberghi a 4 stelle).
- d) Calcolare per ogni coppia (stato, anno) l'incasso totale effettuato nel corso dei giorni festivi.
- e) Calcolare per ogni albergo l'incasso totale effettuato nell'anno 2005 considerando solo le camere dotate di collegamento satellitare e vasca idromassaggio.

2. Progettazione

- 2.1. Progettare il data warehouse necessario per gestire le informazioni relative agli incassi degli alberghi e all'uso delle camere degli alberghi, in modo da soddisfare le richieste descritte al punto 1. Il data warehouse progettato deve permettere di rispondere in modo efficiente a tutte le interrogazioni frequenti proposte al punto 1 (interrogazioni a), b), c), d), e)).

Il data warehouse realizzato deve contenere le informazioni relative agli ultimi due anni. Al fine di una corretta realizzazione del data warehouse sono state fornite anche le seguenti informazioni:

- Numero di alberghi: ~500
- Numero di stati: ~40
- Numero di città: ~400
- Numero di caratteristiche delle camere (numero di letti, dotata di televisore, ..): ~8

- 2.2. Decidere come gestire la dinamicità (variazioni) dei dati all'interno delle dimensioni.

- 2.3. Rispondere alle interrogazioni frequenti (c) e (e) proposte nel punto 1 utilizzando il linguaggio SQL esteso.

- 2.4. Considerando le caratteristiche del data warehouse realizzato e la cardinalità dei dati memorizzati nel data warehouse, decidere se e quali viste materializzate o indici potrebbe essere utile definite al fine di ottimizzare i tempi di risposta delle interrogazioni proposte al punto 1 (considerare tutte le interrogazioni proposte e non solo quelle risolte in SQL). Motivare le scelte fatte.

Progettazione concettuale

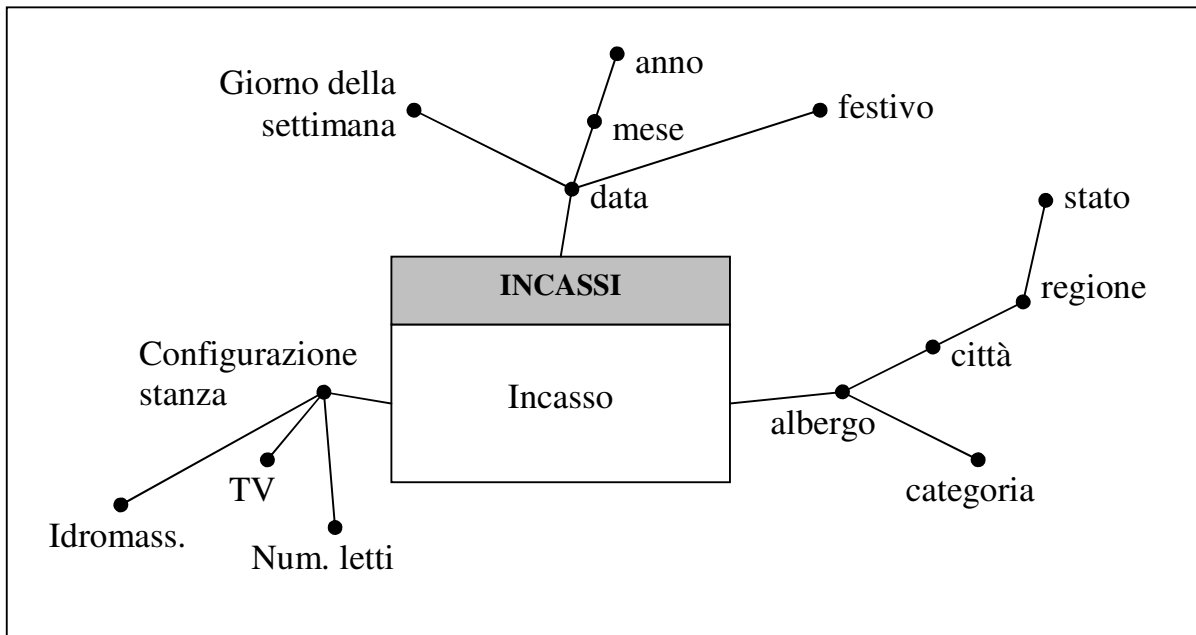


Figura 1 – Fatto Incassi

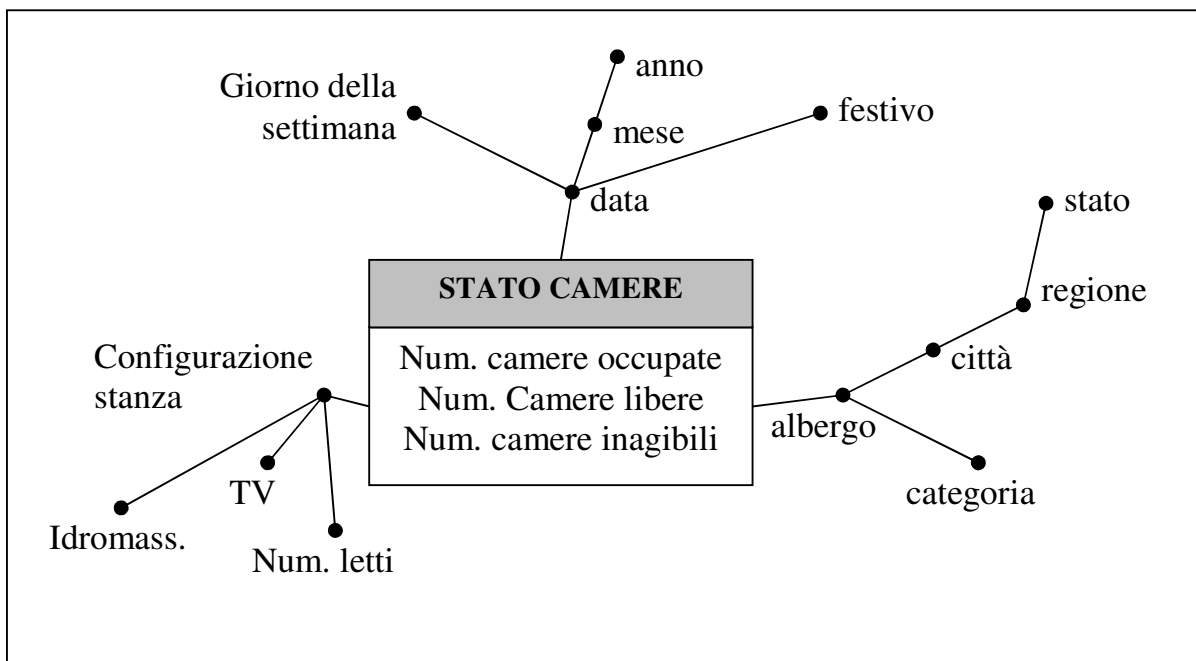


Figura 2 – Fatto Camere

Progettazione logica

Le dimensioni dei due fatti sono tutte identiche, quindi creo una sola tabella per ogni dimensione.

- Dimensioni

ALBERGO (CodA, Albergo, Categoria, Città, Regione, Stato)

TEMPO (CodT, Data, Mese, Anno, Giorno_settimana, Festivo)

CONFIGURAZIONE_STANZA (CodC, Num_letti, TV, Vasca_idromassaggio)

I due fatti sono simili (stesse dimensioni con lo stesso livello di dettaglio) posso quindi creare una sola tabella dei fatti con tutte le misure.

- Fatti

GESTIONE_CAMERE (CodA, CodT, CodC, Incasso, Num_cam_occup, Num_cam_libere, Num_cam_inag);

Gesione dinamicità delle dimensioni

Dimensione ALBERGO

- Cambia la città, la regione o lo stato in cui si trova l'albergo.
E' un evento molto raro. Se succede, posso usare lo scenario temporale TIPO1 (*oggi per ieri*)
- Cambia la categoria dell'albergo
 - Uso TIPO1 (*oggi per ieri*) se mi interessa l'albergo (sovrascrivo la nuova categoria e pertanto tutti gli eventi, anche quelli passati, verranno interpretati in base all'attuale configurazione)
 - Uso TIPO2 (*oggi o ieri*) se mi interessa fare analisi precise in base alla categoria (il cambio di categoria porta alla "generazione" di un nuovo albergo/tupla nella dimensione ALBERGO)

Dimensione CONFIGURAZIONE_STANZA

- In caso di aggiunta di una nuova caratteristica per le camere aggiungo un nuovo attributo alla dimensione.

Query

- E) 2. Calcolare per ogni albergo l'incasso totale nell'anno 2005 considerando solo le camere dotate di TV e di vasca idromassaggio.

```
SELECT A.Albergo, SUM(Incasso)
FROM GESTIONE_CAMERE F, TEMPO T, ALBERGO A,
CONFIGURAZIONE_STANZA C
WHERE F.CodT = T.CodT AND
      F.CodA = A.CodA AND
      F.CodC = C.CodC AND
      T.anno=2005 AND
      C.TV='si' AND
      C.Vasca_idromassaggio='si'
```


GROUP BY A.CodA, A.Albergo

3. Trovare per ogni regione il numero medio giornaliero di camere libere negli alberghi a 5 stelle del mese di gennaio 2007.

```
SELECT A.Regione, SUM(F.Num_cam_libere)/COUNT(distinct T.data)
FROM ALBERGO A, GESTIONE_CAMERE F, TEMPO T
WHERE F.CodA=A.CodA AND
      F.CodT=T.CodT AND
      A.Categoria='5 stelle' AND
      T.Mese='Gennaio 2007'
GROUP BY A.Regione
```

4. Trovare le città con almeno 2 alberghi nelle quali, il 10 marzo 2007, tutte le camere di tutti gli alberghi erano agibili.

```
SELECT A.citta
FROM ALBERGO A, GESTIONE_CAMERE F, TEMPO T
WHERE F.CodA=A.CodA AND
      F.CodT=T.CodT AND
      T.data='10/03/07' AND
      A.citta IN
      (
        SELECT A2.Citta
        FROM ALBERGO A2
        GROUP BY A2.Citta
        HAVING COUNT(*)>1
      )
GROUP BY A.citta
HAVING SUM(num_cam_inagib)=0
```

5. Selezionare le regioni in cui non ci sono alberghi a 3 stelle

```
SELECT A.Regione
FROM ALBERGO A
WHERE A.Regione NOT IN
(
  SELECT A2.Regione
  FROM ALBERGO A2
  WHERE A2.Categoria='3 stelle'
)
```

- C) 6. Relativamente all'anno 2005, selezionare per ogni coppia (stato, mese) l'incasso mensile calcolato considerando solo gli alberghi a 4 stelle e l'incasso cumulativo da inizio anno (sempre considerando solo gli alberghi a 4 stelle).

```
SELECT stato, mese, SUM(Incasso),
      SUM(SUM(Incasso)) OVER (PARTITION BY stato
                              ORDER BY mese
```

```
                                ROWS UNBOUNDED PRECEDING)
FROM GESTIONE_CAMERE F, TEMPO T, ALBERGO A
WHERE F.CodT = T.CodT AND
      F.CodA = A.CodA AND
      A.Categoria='4 stelle' AND
      T.Anno=2005
GROUP BY stato, mese;
```

Laboratorio

SQL SERVER 2005 – Integration services: Cosa rappresentano le frecce di collegamento tra i task nel Control Flow e cosa invece nel Data Flow?

Nel Control Flow i collegamenti tra i task definiscono le *precedenze* tra i task da eseguire mentre nel Data Flow i collegamenti rappresentano un *passaggio fisico* di dati tra i task.

Esercizio di progettazione di un data warehouse

Progettare un data warehouse per la gestione delle problematiche illustrate nei punti seguenti relative ai magazzini di una ditta di elettrodomestici italiana con sedi in tutta Italia.

Descrizione del problema

Una grande ditta di elettrodomestici ha magazzini sparsi in tutta Italia. La dirigenza dell'azienda ha la necessità di analizzare l'attuale uso dei magazzini al fine di capire se e quanto i vari magazzini sono effettivamente utilizzati in funzione della loro dislocazione geografica. La finalità dell'azienda è quella di decidere quali magazzini ampliare, quali ridimensionare e quali chiudere o affittare, anche solo parzialmente, a terze parti.

La dirigenza della ditta stipula periodicamente delle polizze assicurative per coprire i danni dei furti che si verificano nei magazzini. I prezzi delle polizze contro i furti sono legati alla quantità media giornaliera di prodotti presenti nei magazzini e al loro valore. Esistono diverse tipologie di polizze e la dirigenza vuole capire quali sono le polizze più adeguate per i propri magazzini analizzando quali modelli di prodotti e in quali quantità sono mediamente presenti a livello giornaliero nei magazzini. Alcune polizze sono specifiche per un singolo magazzino mentre altre sono relative ad insiemi di magazzini che si trovano nella stessa città, regione o provincia. In particolare i premi assicurativi delle polizze sono calcolati considerando, su base media giornaliera:

- il numero di prodotti presenti nei magazzini per ogni modello di prodotto presente
- il valore dei prodotti presenti nei magazzini per ogni modello di prodotti.

Viste le problematiche appena descritte la ditta ha deciso di creare un apposito data warehouse per gestire le informazioni relative all'uso dei magazzini e al loro contenuto.

Relativamente all'analisi dell'uso dei magazzini (superficie utilizzata) la dirigenza è interessata ad analizzare la percentuale giornaliera di superficie libera rispetto alla superficie totale disponibile in ogni magazzino in funzione del magazzino e del luogo in cui esso si trova (città, provincia, regione). La superficie totale disponibile per ogni magazzino può variare di giorno in giorno in quanto alcune aree dei magazzini possono essere affittate a terzi o essere temporaneamente inagibili (queste aree non fanno parte della superficie totale disponibile). L'analisi deve essere possibile a livello di singola data, ma anche a livello di mese, trimestre, quadrimestre, semestre e anno.

Per stimare i costi delle possibili polizze assicurative la ditta deve poter analizzare il numero medio giornaliero di prodotti presenti nei magazzini e il loro valore complessivo giornaliero a livello di singola data, a livello di mese, di trimestre, quadrimestre, di semestre e di anno. Tali informazioni devono essere disponibili in funzione del modello dei prodotti e della loro categoria (scopa elettrica, impastatrice, ..), del magazzino e del luogo in cui si trova (città, provincia, regione).

Sono di seguito riportate **alcune** delle interrogazioni frequenti di interesse per la dirigenza della ditta:

- a) Relativamente al primo trimestre dell'anno 2003, considerando solo i magazzini della città di Torino, trovare per ogni coppia (magazzino,data) il valore complessivo di prodotti presenti in tale data nel magazzino e il valore complessivo medio giornaliero di prodotti presenti nel magazzino nel corso della settimana precedente la data in esame (data in esame inclusa).
- b) Relativamente all'anno 2004, trovare per ogni coppia(città,data) la percentuale di superficie libera giornaliera nella città. Associare ad ogni coppia un attributo di rank legato alla percentuale di superficie libera giornaliera nella città (1 per la coppia con la più bassa percentuale di superficie libera giornaliera).
- c) Relativamente ai primi sei mesi dell'anno 2004, trovare per ogni coppia (magazzino,data) la percentuale di superficie libera giornaliera.
- d) Relativamente all'anno 2003, trovare per ogni coppia (magazzino,mese) il valore complessivo medio giornaliero di prodotti presenti.
- e) Relativamente all'anno 2003, trovare per ogni regione il valore complessivo medio giornaliero di prodotti presenti nella regione.
- f) Relativamente all'anno 2004, trovare per ogni coppia(mese, regione) la percentuale di superficie libera giornaliera nella regione.

Progettazione

1. Progettare il data warehouse necessario per gestire le necessità della ditta di elettrodomestici in modo da soddisfare le richieste descritte nelle specifiche del problema. Il data warehouse progettato deve inoltre permettere di rispondere in modo efficiente a **tutte** le interrogazioni frequenti proposte nelle specifiche del problema.

Il data warehouse realizzato deve contenere le informazioni relative agli ultimi 2 anni. Al fine di una corretta realizzazione del data warehouse sono state fornite le seguenti informazioni:

- Numero di modelli diversi di prodotti: ~100
 - Numero di categorie diverse di prodotti: ~10
 - Numero di magazzini: ~100
 - Numero di magazzini di Torino: ~5
 - Numero di città: ~90
 - Numero di regioni: ~40
 - Numero di province: ~10
2. Esprimere le interrogazioni frequenti (a), (b), (d) delle specifiche del problema utilizzando il linguaggio SQL esteso.
 3. Considerando le caratteristiche del data warehouse realizzato e la cardinalità dei dati memorizzati nel data warehouse, decidere se e quali viste materializzate o indici potrebbe essere utile definire al fine di ottimizzare i tempi di risposta delle interrogazioni proposte nelle specifiche del problema (considerare **tutte** le interrogazioni proposte e non solo quelle risolte in SQL al punto 2). Motivare le scelte fatte.

```
/* #####*
 * Sistemi per la gestione di basi di dati      *
 * Esame del 31-01-2007                        *
 * Clemenza Carmelo - matricola: 147993        *
 * #####*
 */
```

TEMPO(COD_T, DATA, MESE, TRI_M, QUADRI_M, SE_M, ANNO); -> CARD=365*2=730

MAGAZZINO(COD_M, MAGAZZINO, CITTA, PROVINCIA, REGIONE); -> CARD=100

SUPERFICIE(COD_T, COD_M, SUPERFICIE_LIBERA, SUPERFICIE_TOTALE); -> CARD=730*100=73.000

TEMPO, MAGAZZINO -> CONDIVISA CON 'SUPERFICIE'

MODELLO(COD_MOD, MODELLO_PRODOTTO, CATEGORIA); -> CARD=100

PRODOTTI(COD_T, COD_M, COD_MOD, N_PRODOTTI, VALORE_COMPLESSIVO); -> CARD=730*100*100=7,3*10^6

```
--a)
SELECT MAGAZZINO, DATA, SUM(VALORE_COMPLESSIVO) AS VALORE_COMPLESSIVO_PRODOTTI,
       AVG(SUM(VALORE_COMPLESSIVO)) OVER (PARTITION BY COD_M
                                           ORDER BY DATA
                                           RANGE BETWEEN INTERVAL 6 DAYS PRECEDING AND CURRENT
                                           ROW) AS
                                           VALORE_COMPLESSIVO_PRODOTTI_NELLA_SETTIMANA_PRECEDEM
                                           TE
FROM PRODOTTI P, TEMPO T, MAGAZZINO M
WHERE P.COD_T=T.COD_T AND P.COD_M=M.COD_M AND T.ANNO=2003 AND T.TRI_M=1 AND M.CITTA='TORINO'
GROUP BY MAGAZZINO, DATA, COD_M;

--b)
SELECT CITTA, DATA, 100*SUPERFICIE_LIBERA/SUPERFICIE_TOTALE AS
PERCENTUALE_SUPERFICIE_LIBERA_IN_CITTA,
       RANK() OVER (ORDER BY 100*SUPERFICIE_LIBERA/SUPERFICIE_TOTALE) AS POSIZIONE
FROM SUPERFICIE S, TEMPO T, MAGAZZINO M
WHERE S.COD_T=T.COD_T AND S.COD_M=M.COD_M AND T.ANNO=2004
GROUP BY CITTA, DATA;

--c)
SELECT MAGAZZINO, DATA, 100*SUPERFICIE_LIBERA/SUPERFICIE_TOTALE AS
PERCENTUALE_SUPERFICIE_LIBERA,
FROM SUPERFICIE S, TEMPO T, MAGAZZINO M
WHERE S.COD_T=T.COD_T AND S.COD_M=M.COD_M AND T.ANNO=2004 AND T.SE_M=1;

--d')
SELECT MAGAZZINO, MESE, SUM(VALORE_COMPLESSIVO)/COUNT(DISTINCT DATA) AS
VALORE_COMPLESSIVO_MEDIO_GIORNALIERO
FROM PRODOTTI P, TEMPO T, MAGAZZINO M
WHERE P.COD_T=T.COD_T AND P.COD_M=M.COD_M AND T.ANNO=2003
GROUP BY MAGAZZINO, MESE, COD_M;

--d'')
SELECT DISTINCT MAGAZZINO, MESE, AVG(SUM(VALORE_COMPLESSIVO)) OVER (PARTITION BY COD_M, MESE)
AS VALORE_COMPLESSIVO_MEDIO_GIORNALIERO
FROM PRODOTTI P, TEMPO T, MAGAZZINO M
WHERE P.COD_T=T.COD_T AND P.COD_M=M.COD_M AND T.ANNO=2003
GROUP BY MAGAZZINO, MESE, DATA, COD_M;

--e')
SELECT REGIONE, SUM(VALORE_COMPLESSIVO)/COUNT(DISTINCT DATA) AS
VALORE_COMPLESSIVO_MEDIO_GIORNALIERO
FROM PRODOTTI P, TEMPO T, MAGAZZINO M
WHERE P.COD_T=T.COD_T AND P.COD_M=M.COD_M AND T.ANNO=2003
GROUP BY REGIONE;

--e'')
SELECT DISTINCT REGIONE, AVG(SUM(VALORE_COMPLESSIVO)) OVER (PARTITION BY REGIONE) AS
VALORE_COMPLESSIVO_MEDIO_GIORNALIERO
FROM PRODOTTI P, TEMPO T, MAGAZZINO M
WHERE P.COD_T=T.COD_T AND P.COD_M=M.COD_M AND T.ANNO=2003
GROUP BY REGIONE, DATA;
```

--f)

```
SELECT DISTINCT REGIONE, MESE, AVG(SUM(SUPERFICIE_LIBERA)/SUM(SUPERFICIE_TOTALE)) OVER (
PARTITION BY REGIONE, MESE) AS
FROM SUPERFICIE S, TEMPO T, MAGAZZINO M
WHERE S.COD_T=T.COD_T AND S.COD_M=M.COD_M AND T.ANNO=2004
GROUP BY MESE, REGIONE, DATA;
```

--CARDINALITA DELLE INTERROGAZIONI

a) CARDINALITA~= $90 \cdot 5$ =	450	<< $7,3 \cdot 10^6$	-> VISTA MATERIALIZZATA CONVENIENTE
b) CARDINALITA~= $365 \cdot 90$ =	32.950	< 73.000	-> VISTA MATERIALIZZATA NON CONVENIENTE
c) CARDINALITA~= $180 \cdot 100$ =	18.000	< 73.000	-> VISTA MATERIALIZZATA NON CONVENIENTE
d) CARDINALITA~= $12 \cdot 100$ =	1.200	<< $7,3 \cdot 10^6$	-> VISTA MATERIALIZZATA CONVENIENTE
e) CARDINALITA~= 40 =	40	<< $7,3 \cdot 10^6$	-> VISTA MATERIALIZZATA CONVENIENTE
f) CARDINALITA~= $40 \cdot 12$ =	480	<< 73.000	-> VISTA MATERIALIZZATA CONVENIENTE

Esercizio di progettazione di un data warehouse

Progettare un data warehouse per la gestione delle problematiche illustrate nei punti seguenti relative ad un sito per la pubblicazione di annunci relativi all'affitto di immobili.

Descrizione del problema

Il sito internet www.cerca_la_tua_casa.it permette di fare ricerche in tutte le città italiane per trovare gli immobili in affitto. I privati o le agenzie possono pubblicare su questo sito gli annunci relativi agli immobili disponibili, specificando per ogni immobile le sue caratteristiche principali: zona della città in cui si trova (quartiere), tipo di immobile (attico, mansarda, villa a schiera, rustico,...), prezzo di affitto mensile, superficie in metri quadri, numero di stanze, numero di bagni, piano a cui è sito l'immobile, eventuale presenza dell'ascensore o della cantina. In più è possibile segnalare l'eventuale arredamento già presente nell'immobile: presenza di tavolo/sedie, frigorifero, forno, stufa, ventilatore, letto, lavatrice, lavastoviglie, tv, ecc.

Gli annunci vengono aggiornati settimanalmente, ogni lunedì vengono eliminati gli annunci relativi agli appartamenti che sono stati occupati e vengono aggiunti i nuovi annunci relativi agli appartamenti liberi.

Gli utenti del sito possono aggiungere alla sezione personale "aggiungere ai preferiti" tutti gli immobili ai quali sono interessati, in modo da poterli poi visionare in un'unica pagina e poterli confrontare più agevolmente.

La società che gestisce il sito internet è interessata ad analizzare la situazione degli affitti in Italia. I parametri presi in considerazione per questa analisi sono: la disponibilità di immobili in affitto (n° di immobili in affitto), la media del prezzo di affitto per immobile, e la media del prezzo di affitto al metro quadro. Questi parametri si devono poter analizzare in funzione

- della settimana, del mese, del bimestre, del trimestre, del quadrimestre, del semestre e dell'anno
- della zona d'Italia (nord, centro, sud, isole), della regione, della provincia, della città, della zona della città (quartiere) in cui è situato l'immobile
- della presenza di università nella città in cui è situato l'immobile
- del tipo di immobile e delle tipologie di arredi eventualmente presenti (sedia, tavolo, frigorifero, ecc.).
- del numero di stanze

Si è anche interessati a fare delle statistiche in base a quali sono gli immobili "preferiti" dagli utenti, e quindi a valutare il numero medio di utenti interessati ad un immobile in funzione:

- dell'anno e della stagione (primavera, estate, autunno, inverno)
- dalla zona d'Italia (nord, centro, sud, isole), della regione, della provincia, della città, della zona della città (quartiere)
- della presenza di università nella città in cui è situato l'immobile
- del tipo di immobile
- del range di costo dell'affitto (100-200€, 200-300€,...) e del range di dimensione (0-50 mq, 50-100 mq,...) dell'immobile
- del numero di stanze

Sono di seguito riportate **alcune** delle interrogazioni frequenti di interesse per la società che gestisce il sito contenente gli annunci di affitto:

- a) Relativamente al 2004, considerando solo gli immobili situati in città nelle quali sono presenti delle università, trovare per ogni coppia (città, mese) il costo medio di affitto per immobile e il costo medio di affitto per immobile da inizio anno nella città in esame.
- b) Relativamente al mese di settembre 2004, considerando solo gli immobili situati in provincia di Torino, trovare per ogni coppia (città, settimana) il numero totale di immobili disponibili, il rapporto tra il numero totale di immobili disponibili relativi alla coppia in esame e il numero totale di immobili disponibili nella settimana in esame. Associare ad ogni coppia un attributo di rank legato al numero totale di immobili disponibili. Associare il valore 1 alla coppia con il maggior numero di immobili disponibili. Ordinare i dati in funzione dell'attributo di rank.
- c) Relativamente alla stagione estiva del 2005, considerando solo le mansarde dotate di letto, frigorifero e tavolo situate presso la città di Rimini, trovare per ogni coppia (zona città, range affitto) il numero medio di utenti interessati per immobile e il numero medio di utenti interessati per immobile nella zona della città della coppia in esame. Ordinare i dati in funzione della zona della città e del numero medio di utenti interessati per immobile.
- d) Considerando solo gli immobili situati in città nelle quali sono presenti delle università e dotati di letto e tavolo, trovare per ogni tripletta (città, mese, anno) il costo medio di affitto per immobile, il costo

medio di affitto al metro quadro, il costo medio di affitto per immobile da inizio anno nella città in esame.

- e) Considerando i mesi di settembre, ottobre e novembre 2004 e gli immobili situati presso la regione Piemonte, trovare per ogni città il prezzo medio di affitto per immobile e il prezzo medio di affitto per immobile relativo alla provincia della città in esame.
- f) Relativamente al 2004, considerando solo gli immobili situati in città nelle quali sono presenti delle università e dotati di letto e tavolo, trovare per ogni coppia (città, mese) il costo medio di affitto per immobile e il costo medio di affitto al metro quadro.

Progettazione

1. Progettare il data warehouse necessario per gestire le necessità della società che gestisce il sito contenente gli annunci di affitto in modo da soddisfare le richieste descritte nelle specifiche del problema. Il data warehouse progettato deve inoltre permettere di rispondere in modo efficiente a **tutte** le interrogazioni frequenti proposte nelle specifiche del problema.
Il data warehouse realizzato deve contenere le informazioni relative agli ultimi 2 anni. Al fine di una corretta realizzazione del data warehouse sono state fornite le seguenti informazioni:
 - Numero di zone d'Italia ~ 4
 - Numero di provincie ~ 100
 - Numero di città ~ 8000
 - Numero di zone città ~ 10000
 - Numero di città presso cui ci sono delle università ~ 100
 - Numero di tipologie di immobili ~ 5
 - Numero di tipologie di arredi ~ 10
 - Numero di stanze ~ da 1 a 5
 - Numero di range di costo d'affitto ~ 10
 - Numero di range di dimensioni immobili ~ 10
2. Esprimere le interrogazioni frequenti (b), (d), (e) delle specifiche del problema utilizzando il linguaggio SQL esteso.
3. Considerando le caratteristiche del data warehouse realizzato e la cardinalità dei dati memorizzati nel data warehouse, decidere se e quali viste materializzate o indici potrebbe essere utile definire al fine di ottimizzare i tempi di risposta delle interrogazioni proposte nelle specifiche del problema (considerare **tutte** le interrogazioni proposte e non solo quelle risolte in SQL al punto 2). Motivare le scelte fatte.

```

/* #####*
 * Sistemi per la gestione di basi di dati *
 * Esame del 24-04-2008 *
 * Clemenza Carmelo - matricola: 147993 *
 * #####*
 */

```

NOTE: N_STANZE, TIPO_IMMOBILE -> DIMENSIONI DEGENERI -> JUNK DIMENSION
 ARREDO -> DIMENSIONE FORMATA DA 10 ATTRIBUTI BOOLEANI LINEARMENTE INDIPENDENTI -> UNICA
 TABELLA

```

TEMPO(COD_T, SETTIMANA, MESE, BI_M, TRI_M, QUADRI_M, SE_M, ANNO); -> CARD=52*2=104
QUARTIERE(COD_Q, QUARTIERE, CITTA, PROVINCIA, REGIONE, ZONA_ITALIA, UNIVERSITA); -> CARD=10.000
IMMOBILE(COD_I, N_STANZE, TIPO_IMMOBILE); -> CARD~10*5=50
ARREDO(COD_A, TAVOLO, SEDIE, FRIGORIFERO, FORNO, STUFA, VENTILATORE, LETTO, LAVATRICE,
LAVASTOVIGLIE, TV); -> CARD=2^10=1.024
AFFITTI(COD_T, COD_Q, COD_I, COD_A, N_PREFERITI, SUPERFICIE_COMPLESSIVA, AFFITTO_COMPLESSIVO
); -> CARD=104*10.000*50*1.024=5*10^10

```

```

QUARTIERE, IMMOBILE, ARREDO -> CONDIVISA CON 'AFFITTI'
STAGIONE(COD_S, STAGIONE, ANNO); -> CARD=4*2=8
RANGE_DIMENSIONE_AFFITTO(COD_R, RANGE_DIMENSIONE, RANGE_AFFITTO); -> CARD=10*10=100
PREFERITI(COD_Q, COD_I, COD_A, COD_S, COD_R, N_IMMOBILI, N_UTENTI); -> CARD=10.000*50*1.024*8
*100=4*10^11

```

```
--a)
SELECT CITTA, MESE,
       SUM(AFFITTO_COMPLESSIVO)/SUM(N_PREFERITI) AS COSTO_MEDIO_AFFITTO_PER_IMMOBILE,
       SUM(SUM(AFFITTO_COMPLESSIVO))/SUM(SUM(N_PREFERITI)) OVER (PARTITION BY CITTA
                                                                    ORDER BY MESE
                                                                    ROWS UNBOUNDED PRECEDING) AS
                                                                    COSTO_MEDIO_AFFITTO_PER_IMMOB
                                                                    ILE_DA_INIZIO_ANNO

FROM AFFITTI A, QUARTIERE Q, TEMPO T, IMMOBILE I
WHERE A.COD_T=T.COD_T AND A.COD_Q=Q.COD_Q AND A.COD_I=I.COD_I AND T.ANNO=2004 AND Q.
UNIVERSITA=TRUE
GROUP BY CITTA, MESE;

--b)
SELECT SETTIMANA, CITTA, TOTALE_PREFERITI_DISPONIBILI,
       TOTALE_PREFERITI_DISPONIBILI/TOTALE_PREFERITI_PER_SETTIMANA
FROM (SELECT SETTIMANA, CITTA, SUM(N_PREFERITI) AS TOTALE_PREFERITI_DISPONIBILI,
          SUM(SUM(N_PREFERITI)) OVER (PARTITION BY SETTIMANA) AS
          TOTALE_PREFERITI_PER_SETTIMANA
FROM AFFITTI A, QUARTIERE Q, TEMPO T
WHERE A.COD_Q=Q.COD_Q AND A.COD_T=T.COD_T AND T.MESE=9 AND T.ANNO=2004 AND Q.PROVINCIA=
'TORINO'
GROUP BY SETTIMANA, CITTA);

--c)
SELECT QUARTIERE, RANGE_AFFITTO,
       SUM(N_UTENTI_INTERESSATI)/SUM(N_PREFERITI) AS UTENTI_INTERESSATI_PER_IMMOBILE,
       SUM(SUM(N_UTENTI_INTERESSATI))/SUM(SUM(N_PREFERITI)) OVER (PARTITION BY QUARTIERE) AS
       UTENTI_INTERESSATI_PER_IMMOBILE_NEL_QUARTIERE
FROM PREFERITI P, QUARTIERE Q, IMMOBILE, RANGE_DIMENSIONE_AFFITTO R, STAGIONE S, ARREDO ARR
WHERE P.COD_Q=Q.COD_Q AND P.COD_I=IMMOBILE.COD_I AND P.COD_R=R.COD_R AND P.COD_S=S.COD_S AND
P.COD_A=ARR.COD_A
      STAGIONE='ESTATE' AND ANNO=2005 AND TIPO_IMMOBILE='MANSARDE' AND CITTA='RIMINI' AND
      LETTO=TRUE AND FRIGORIFERO=TRUE AND TAVOLO=TRUE
GROUP BY QUARTIERE, RANGE_AFFITTO, COD_Q
ORDER BY QUARTIERE, UTENTI_INTERESSATI_PER_IMMOBILE DESC;

--d)
SELECT CITTA, ANNO, MESE,
       SUM(AFFITTO_COMPLESSIVO)/SUM(N_PREFERITI) AS COSTO_MEDIO_AFFITTO_PER_IMMOBILE,
       SUM(AFFITTO_COMPLESSIVO)/SUM(METRI_QUADRI_COMPLESSIVI) AS
       COSTO_MEDIO_AFFITTO_PER_METRO_QUADRO,
       SUM(SUM(AFFITTO_COMPLESSIVO))/SUM(SUM(N_PREFERITI)) OVER (PARTITION BY CITTA, ANNO
                                                                    ORDER BY MESE
                                                                    ROWS UNBOUNDED PRECEDING) AS
                                                                    COSTO_MEDIO_AFFITTO_PER_IMMOBI
                                                                    LE_DA_INIZIO_ANNO

FROM AFFITTI A, QUARTIERE Q, TEMPO T, IMMOBILE, ARREDO ARR
WHERE A.COD_Q=Q.COD_Q AND A.COD_T=T.COD_T AND A.COD_I=IMMOBILE.COD_I AND
      A.COD_A=ARR.COD_A AND ARR.LETTO=TRUE AND ARR.TAVOLO=TRUE
GROUP BY CITTA, ANNO, MESE;
```

```
--e)
```

```
SELECT CITTA, SUM(AFFITTO_COMPLESSIVO)/SUM(N_PREFERITI) AS COSTO_MEDIO_AFFITTO_PER_IMMOBILE,
       SUM(SUM(AFFITTO_COMPLESSIVO))/SUM(SUM(N_PREFERITI)) OVER (PARTITION BY PROVINCIA) AS
       COSTO_MEDIO_AFFITTO_PER_IMMOBILE_SU_PROVINCIA
FROM AFFITTI A, QUARTIERE Q, TEMPO T
WHERE A.COD_Q=Q.COD_Q AND A.COD_T=T.COD_T AND T.MESE>=9 AND T.MESE<=11 AND T.ANNO=2004 AND Q.
REGIONE='PIEMONTE'
GROUP BY PROVINCIA, CITTA;
```

```
--f)
```

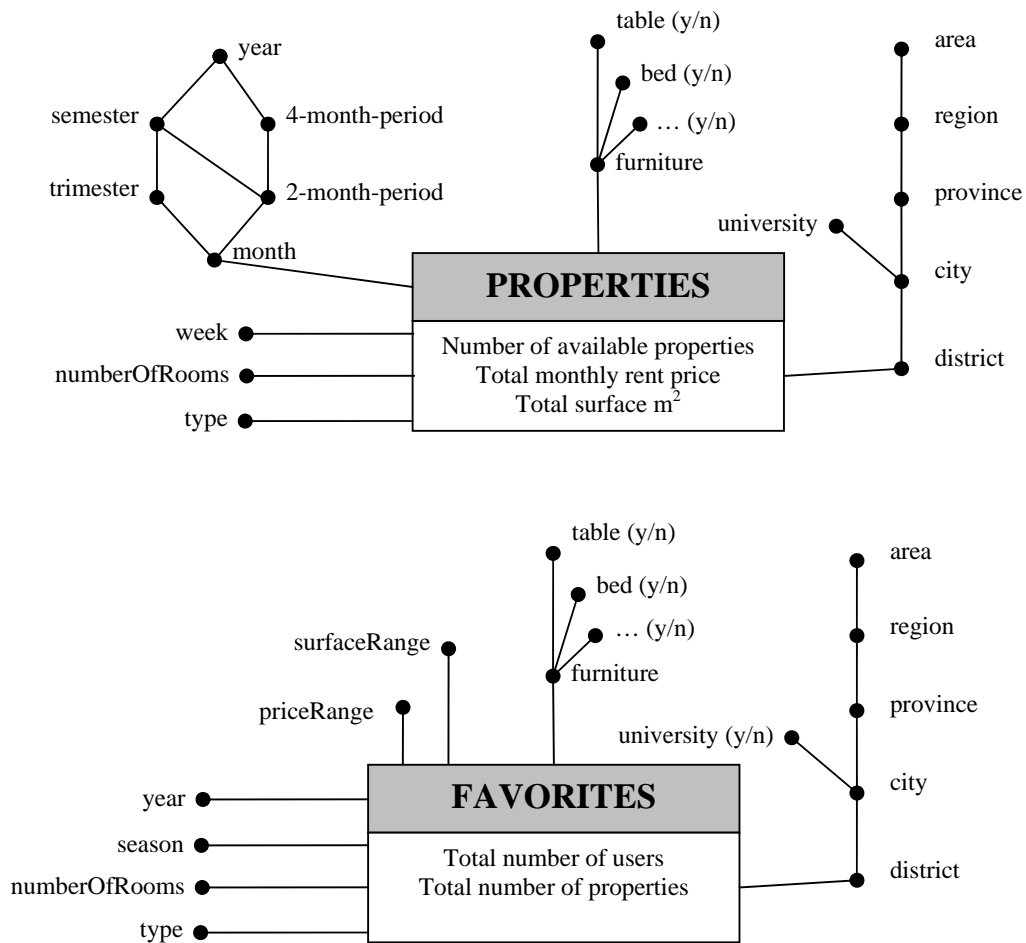
```
SELECT CITTA, MESE,
       SUM(AFFITTO_COMPLESSIVO)/SUM(N_PREFERITI) AS COSTO_MEDIO_AFFITTO_PER_IMMOBILE,
       SUM(AFFITTO_COMPLESSIVO)/SUM(METRI_QUADRI_COMPLESSIVI) AS
       COSTO_MEDIO_AFFITTO_PER_METRO_QUADRO
FROM AFFITTI A, QUARTIERE Q, TEMPO T, IMMOBILE, ARREDO ARR
WHERE A.COD_Q=Q.COD_Q AND A.COD_T=T.COD_T AND A.COD_I=IMMOBILE.COD_I AND A.COD_A=ARR.COD_A AND
       T.ANNO=2004 AND Q.UNIVERSITA=TRUE AND ARR.LETTO=TRUE AND ARR.TAVOLO=TRUE
GROUP BY CITTA, MESE;
```

Analisi di basi di dati

Politecnico di Torino
 III Facoltà di Ingegneria
 Laurea Specialistica in Ingegneria Informatica

Esame del 07-09-2007 – Soluzione **DRAFT**

Modello Concettuale



Modello Logico

Primary keys are underlined.

Facts

PROPERTIES (monthID, weekID, typeID, roomsID, furnitureID, locationID, numProperties, totPrice, totSurface)

FAVORITES (yearID, seasonID, typeID, roomsID, furnitureID, locationID, surfaceRangeID, priceRangeID, numUsers, numProperties)

Dimensions

WEEK (weekID, week)

MONTH (monthID, month, 2m-period, trimester, 4m-period, semester, year)

TYPE (typeID, type)

ROOMS (roomsID, numberOfRooms)

FURNITURE (furnitureID, table, bed, ...)

LOCATION (locationID, district, city, university, province, region, area)

SEASON (seasonID, season)

YEAR (yearID, year)

PRICE_RANGE (priceID, priceMin, priceMax)

SURFACE_RANGE (surfaceID, surfaceMin, surfaceMax)

Some dimensions could have been directly stored into the fact table, such as the Room dimension.

→ only for Properties fact

→ only for Properties fact

→ shared both facts

→ shared both facts

→ shared both facts

→ shared both facts

→ only for Favorites fact

→ only for Favorites fact

→ only for Favorites fact

→ only for Favorites fact

Since this is a draft, some tables and columns have the same names, but keep in mind that this is discouraged to avoid confusions.

Query A

```
select
  city, month, sum(totPrice)/sum(numProperties),
  ( sum(sum(totPrice)) / sum(sum(numProperties)) ) over (partition by city order by month rows unbounded preceding)
from
  properties p, location l, month m
where
  p.locationID=l.locationID and p.monthID=m.monthID and
  year=2004 and university='y'
group by
  city, month;
```

Query B

```
select
  city, week, sum(numProperties),
  sum(numProperties) / ( sum(sum(numProperties)) over (partition by week) ),
  rank() over (order by sum(numProperties) desc) as position
from
  properties p, location l, month m, week w
where
  p.locationID=l.locationID and p.monthID=m.monthID and p.weekID=w.weekID and
  year=2004 and month='September' and province='Turin'
group by
  city, week
order by
  position;
```

Query C

```
select
  district, surfaceMin, surfaceMax, sum(numUsers) / sum(numProperties) as avgInterestedUsers,
  ( sum(sum(numUsers)) / sum(sum(numProperties)) ) over (partition by district)
from
  favorites f, location l, season s, year y, furniture f, type t, price_range pr
where
  ...JOINS... and season='summer' and year=2005 and type='attic' and city='Rome' and bed='y' and fridge='y' and table='y'
group by
  district, surfaceMin, surfaceMax
order by
  district, avgInterestedUsers;
```

Query D

```
select
    city, month, year,
    sum(totPrice) / sum(numProperties),
    sum(totPrice) / sum(totSurface),
    ( sum(sum(totPrice)) / sum(sum(numProperties)) ) over (partition by city, year order by month rows unbounded preceding)
from
    properties p, location l, month m, furniture f
where
    ...JOINS... and
    bed='y' and table='y' and university='y'
group by
    city, month, year
```

Query E

```
select
    city, sum(totPrice) / sum(numProperties),
    ( sum(sum(totPrice)) / sum(sum(numProperties)) ) over (partition by province)
from
    properties p, location l, month m
where
    ...JOINS... and year=2004 and month>=9 and month<=11 and region='Piedmont'
group by
    city
```

Query F

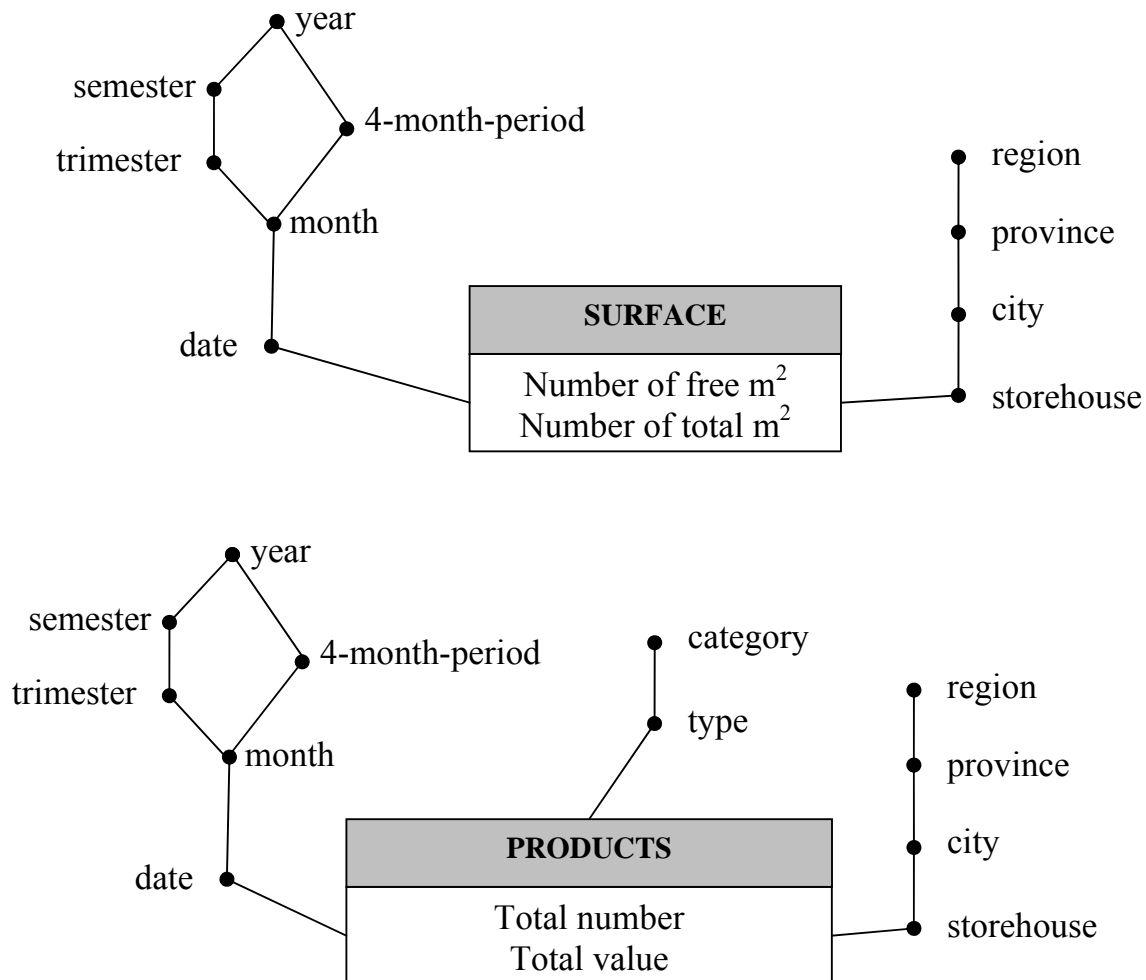
```
select
    city, month,
    sum(totPrice) / sum(numProperties),
    sum(totPrice) / sum(totSurface),
from
    properties p, location l, month m, furniture f
where
    ...JOINS... and year=2004 and university='y' and bed='y' and table='y'
group by
    city, month
```

Analisi di basi di dati

Politecnico di Torino
III Facoltà di Ingegneria
Laurea Specialistica in Ingegneria Informatica

ESAME DEL 31-01-2007 – Soluzione DRAFT

Modello Concettuale



Modello Logico

Primary keys are underlined.

Facts

SURFACE (storehouseID, timeID, m2free, m2tot)

PRODUCTS (storehouseID, timeID, typeID, totNumber, totValue)

Dimensions

TIME (timeID, date, month, trimester, 4month-period, semester, year)

→ shared both facts

TYPES (typeID, type, category)

→ only for Products fact

STOREHOUSES (storehouseID, storehouse, city, province, region)

→ shared both facts

Query A

```
select
  storehouse, date, sum(totValue),
  avg( sum(totValue) ) over (partition by storehouse order by date range between interval '6' day preceding and current row)
from
  products p, storehouses sh, time t
where
  p.storehouseID=sh.storehouseID and p.timeID=t.timeID and
  t.year=2003 and t.trimester=1 and sh.city='Turin'
group by
  storehouseID, storehouse, date;
```

Card: $5 \times (30 \times 3) = 450 \ll 7300k$ → a materialized view on this query is convenient.

Removing the constraints on trimester and city, the view would be useful to answer query **d** and **e** too.

NB: averaging the daily total value over the last week could be done using the $sum(sum(totValue)/7)$ expression, which handles missing days as if their *totValue* were 0, while the proposed solution fills missing values with the week average; furthermore note that *totValue* is a level measure, thus there should be no missing values in the data warehouse.

Query B

```
select
  city, date,
  sum(m2free)/sum(m2tot)*100,
  rank() over (order by sum(m2free)/sum(m2tot) asc)
from
  surface s, storehouses sh, time t
where
  s.storehouseID=sh.storehouseID and s.timeID=t.timeID and t.year=2004
group by
  city, date;
```

Card: $90 \times 365 = 32850 \approx 73000$ → a materialized view on this query is NOT convenient.

Query C

```
select
  storehouse, date, m2free/m2tot,
from
  products p, storehouses sh, time t
where
  p.storehouseID=sh.storehouseID and p.timeID=t.timeID and
  t.year=2004 and t.month>=1 and t.month<=6
group by
  storehouseID, storehouse, date;
```

Card: $100 \times (30 \times 6) = 18000 \approx 73000$ → a materialized view on this query is NOT convenient.

Query D

```
select
    storehouse, month,
    sum(totValue)/count(distinct date)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2003
group by
    storehouseID, storehouse, month;
```

```
select distinct
    storehouse, month,
    avg( sum(totValue) ) over (partition by storehouse, month)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2003
group by
    storehouseID, storehouse, date, month;
```

Card: $100 \times 12 = 1200 \ll 7300k \rightarrow$ a materialized view on this query is convenient and it helps to answer query **e** too.

NB: the DISTINCT command does **not** remove rows with the same storehouse; it removes duplicate rows considering all attribute values of each row.

Query E

```
select
    region, sum(totValue)/count(distinct date)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2003
group by
    region;
```

```
select distinct
    region, avg(sum(totValue)) over (partition by region)
from
    products p, storehouses sh, time t
where
    p.storehouseID=sh.storehouseID and p.timeID=t.timeID and t.year=2003
group by
    region, date;
```

Card: $40 \ll 7300k \rightarrow$ a materialized view on this query is convenient.

Query F

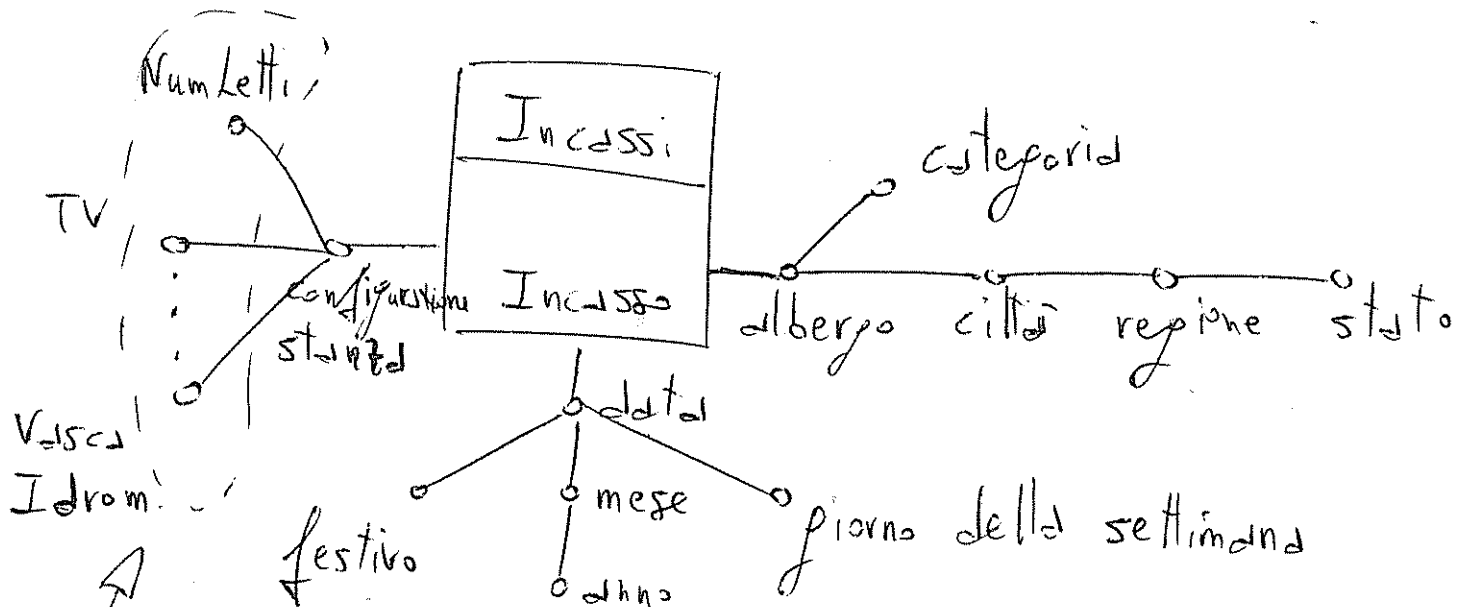
```
select distinct
    region, month,
    avg(sum(m2free)/sum(m2tot)*100) over (partition by region, month)
from
    surface s, storehouses sh, time t
where
    s.storehouseID=sh.storehouseID and s.timeID=t.timeID and t.year=2004
group by
    region, month, date;
```

Card: $40 \times 12 = 480 \ll 7300k \rightarrow$ a materialized view on this query is convenient.

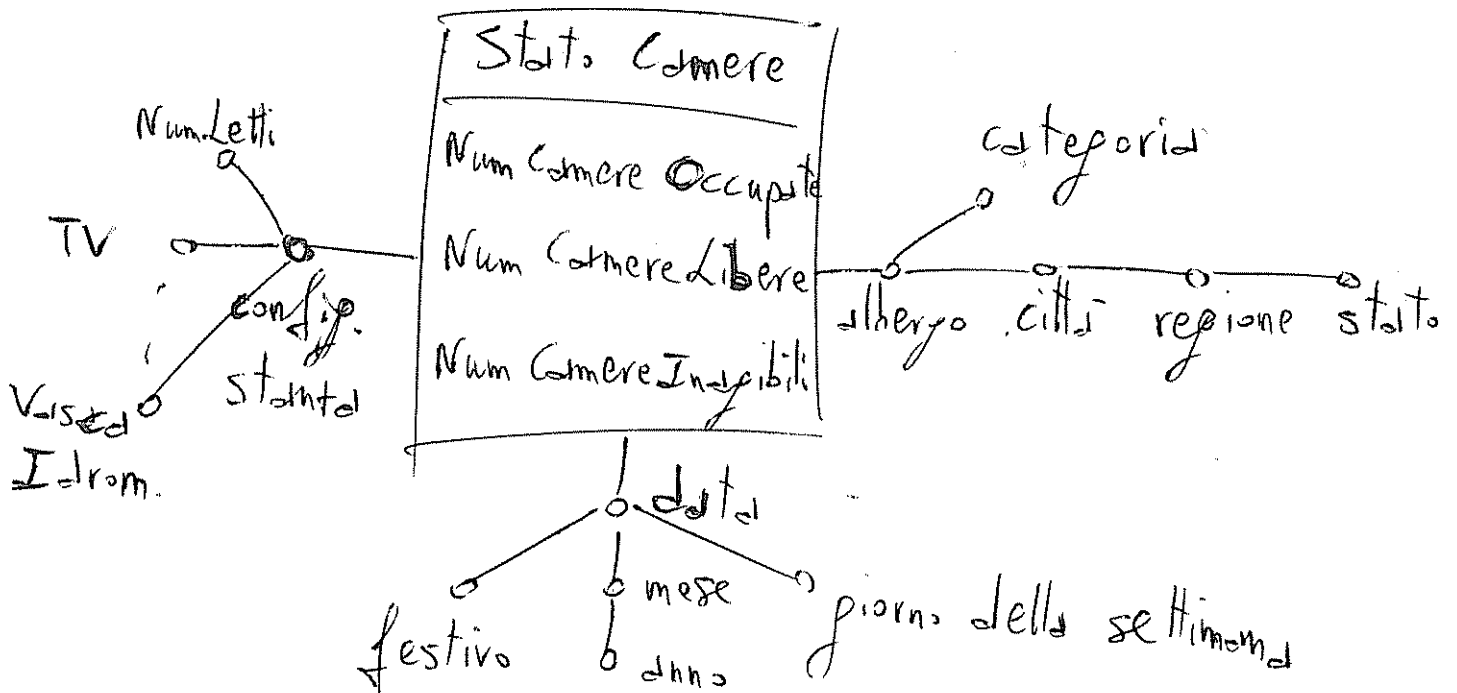
2/2/06

①

2.1 Progetto Concettuale



Assumo i valori si/no. Un attributo x ogni caratteristica delle camere



Progetto
Livello logico

(2)

- Le dimensioni dei 2 fatti sono tutte identiche.

creo una sola tabella per ogni dimensione. Le
tabelle saranno condivise tra i due fatti.

- I due fatti sono simili (stesse dimensioni e
stesso livello di dettaglio) e hanno poche misure

↳ Possibile fare una sola tabella dei fatti
~~che~~ caratterizzata ~~da tutti i fatti~~ ~~da tutte le misure~~
dalle misure di entrambi i fatti

Albergo (CodA, categoria, città, regione, stato, albergo)

Tempo (CodT, data, mese, anno, giorno - sett, festivo)

~~Configurazione Stanza~~ Configurazione Stanza (CodC, NumLetti, TV, ..., VascaIdro)

Fatti (CodA, CodT, CodC, Incasso, Num Camere Occ,
Num Camere Libere, Num Camere Inagibili)

- Dimensione albergo

- Cambia la città, la regione o lo stato in cui si trova un albergo. È un evento molto raro. Se succede posso usare lo scenario temporale oggi x ieri (tipo I)

- Cambia la categoria dell'albergo

- Uso oggi x ieri se mi interessa l'albergo, ~~arrivando~~ (mi porto dietro le vecchie "tuple")

- Uso oggi o ieri se mi interessa fare analisi precise in base alla categoria

(Il cambio di categoria porta alla "generazione" di un nuovo albergo / tuple nella dim. albergo)

Dim. configuratione stanza

Qui non c'è nulla che cambia. Al massimo si aggiunge una nuova caratteristica per le camere. In tal caso si aggiunge un nuovo attributo per la dimensione config. stanza.

2.3

(c) Select stato, mese, SUM(Inciasso),
 SUM(SUM(Inciasso)) OVER (PARTITION BY stato
 ORDER BY mese ROWS UNBOUNDED
 PRECEDING)

FROM FATTI F, TEMPO T, ~~ALBERGO~~ ALBERGO A
 WHERE F.CodT = T.CodT
 AND F.CodA = A.CodA
 AND A.Categoria = '4stelle'
 AND T.Anno = 2005
 GROUP BY stato, mese;

(6)

(e) Select A.Albergo, SUM(Ingresso)
FROM Fatti F, Tempo T, Albergo A,
ConfigurazioneStanza C

WHERE F.CodT = T.CodT

AND ~~AND~~ F.CodA = A.CodA

AND F.CodC = C.CodC

AND T.Anno = 2005

AND C.CollegamentoSatellite = 'si'

AND C.VascaIdrom = 'si'

~~GROUP BY~~ GROUP BY A.CodA, A.Albergo;

2.4 Viste

(7)

Stima cardinalità tabella dei fatti

- Fatti

$$\text{Card}(\text{Fatti}) = 500 \times (2 \times 365) \times 2^8 \approx 93 \times 10^6$$

\uparrow Alberghi \uparrow 2 Anni \uparrow Config. Camere

Interrogazioni

stati mesi 2005

↓ ↓

(a) Cardinalità risultato = 40×12

$40 \times 12 \ll \text{Card}(\text{Fatti}) \Rightarrow$ Può essere utile definire una vista materializzata associata all'interrogazione (a).

Negli definire però una vista ^{materializzata} più generale in cui non ho nella where la condizione sull'anno. L'anno lo uso come attributo di raggruppamento

Vista che seleziona $\text{SUM}(\text{Num Camere Libere})$,
 $\text{SUM}(\text{Num Camere Occ})$,
 $\text{SUM}(\text{Num Camere Indisponibili})$ →

→ e raggruppo per (stato, mese, anno)

Cardinalità massima pari a

$$\begin{array}{ccc} 40 \times 12 \times 2 \\ \uparrow \quad \uparrow \quad \uparrow \\ \text{stati} \quad \text{mesi} \quad \text{anni.} \end{array}$$

(b) Posso usare la vista materializzata definita per l'interrogazione (a). Aggregando i dati in modo opportuno (x stato) ottengo il risultato della query (b).

(c) Cardinalità risultato = 40×12 ~~anni~~
 $\uparrow \quad \uparrow$
 stati mesi 2005
 categorie
 prodotti

$$40 \times 12 \ll \text{Card}(\text{Fatti})$$

↳ Utile definire una vista materializzata in cui raggruppo per (stato, mese) e nella where seleziono in base all'anno (2005) e alla categoria (4 stelle)

Anche in questo caso conviene definire una vista mat. più generica che permette di rispondere a più interrogazioni (non solo alla c)).

Vista mat: → - group by (stato, mese, anno, categoria)

↙ - select SUM(Incasto)
- nulla nella where

$$\text{Card}(Vista\ mat.) = 40 \times 12 \times 2 \times 7$$

Num. categorie alberghi stimato

Raggruppando ^{e selezionando} in modo appropriato posso rispondere in modo efficiente alla query (c) usando la vista materializzata.

(d) Cardinalità risultato = $40 \times 2 \ll \text{Card}(fatti)$

Utile una vista materializzata

- group by (stato, anno)
- where "festivo"
- select SUM(Incasto)

Neptio una vista più generale in cui
raggruppo per (stato, anno, festivo). La cardinalità
passa da 40×2 a $40 \times 2 \times 2$ (differenza
minima).

↓
Vista mat. - group by (stato, anno, festivo)
- where nessuna condizione di
selezione
- select SUM(Ingresso)

(e) Card. risultato = 500 • << Card(Fatti)
 ↑
 Albergo

Conviene generare una vista mat. in
cui raggruppo non solo per albergo, bensì anche
per anno, satellite, vasc. idromassaggio.
La cardinalità aumenta di poco ma posso
rispondere a più interrogazioni.

Card. 500 × 2 × 2 × 2 } - group by (albergo, anno, satellite, vasc. idro)
 - nulla nella where
 - select SUM(Ingresso)

- fatti
- tempo

trova in fretta le tuple della tabella dei fatti relative al 2005 (utile come blocco per risolvere velocemente l'interrogazione).

(b) Utile to stress indite discuss prime.

(C) Utile definire ~~un~~
 - un indice bitmap sulla categoria degli
 alberghi
 - un indice bitmap sull'anno

- un indice bitmap join index relativo alle tabelle:

- fatti

- tempo

~~albergo~~

- un indice bitmap join index relativo alle tabelle:

- fatti

- ~~albergo~~

(d) Indice bitmap su giorno festivo, più indice bitmap index join ~~relativo~~ relativo alle tabelle:

- fatti

- tempo

(13)

(e) Un ~~indice~~ bitmap su collegamento-satellitare, ~~o~~
uno su vettore-idromassaggio, uno sull'anno e

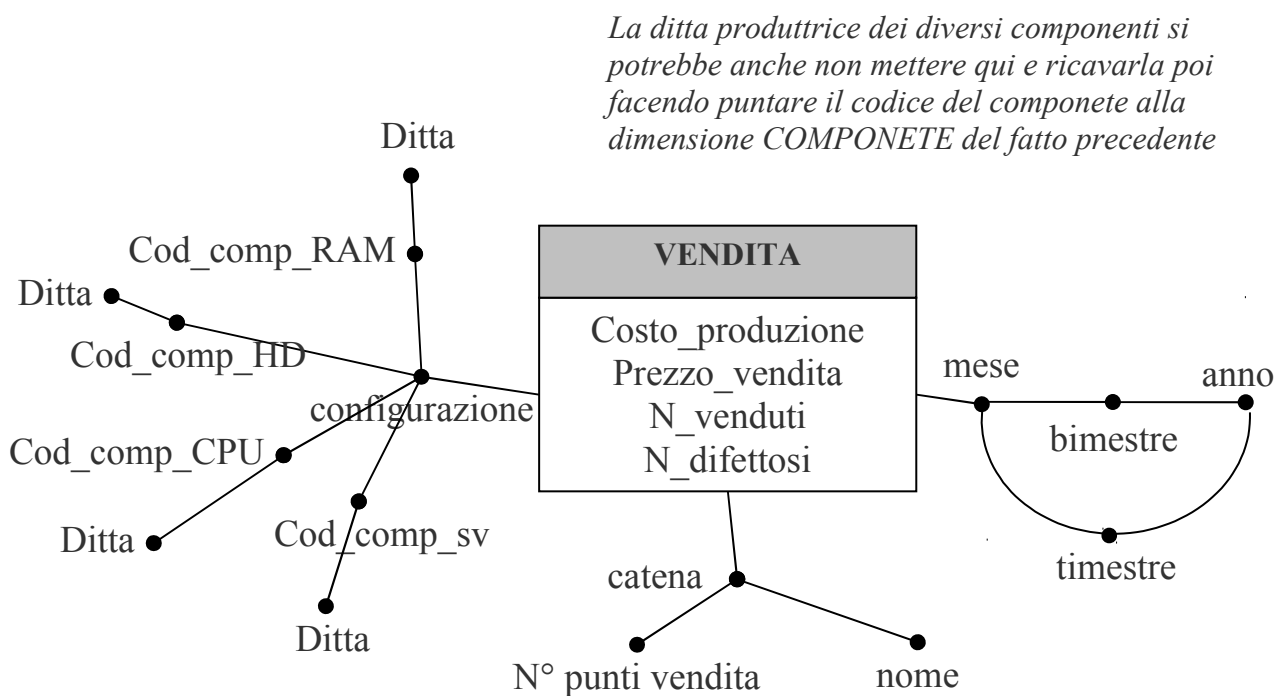
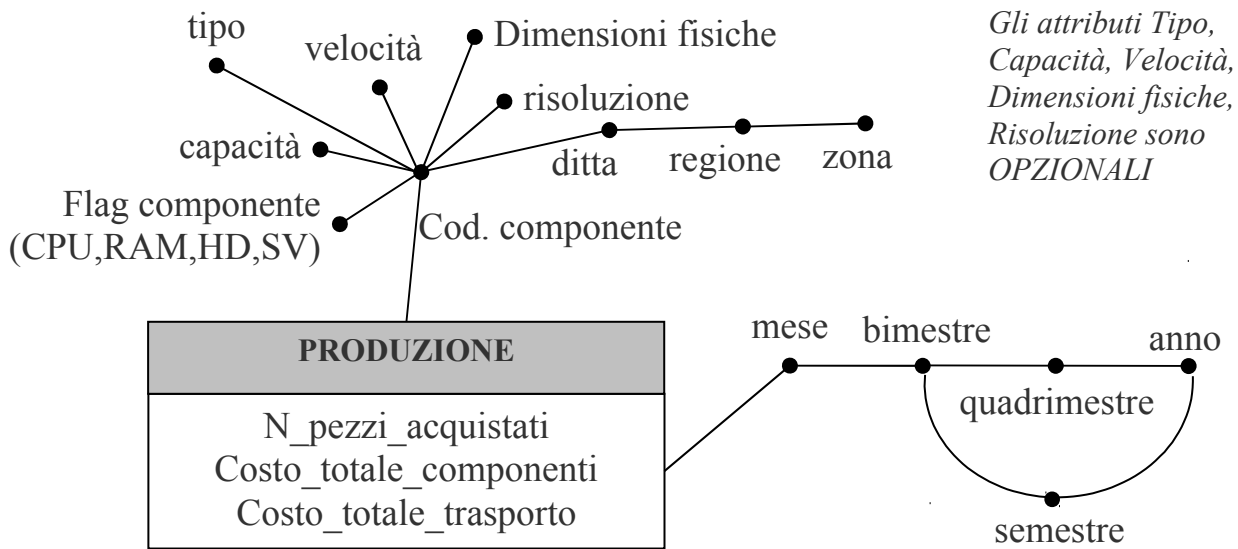
- un bitmap ~~in~~ join index relativo
alle tabelle

- fatti
- tempo

- un bitmap join index relativo alle tabelle

- fatti
- configuration standard

Progettazione concettuale



Fatti:

PRODUZIONE

COMPONENTE (CodC, cod_componente, flag_componente, tipo, capacità, velocità, dimensioni, risoluzione, ditta, regione, zona)

TEMPO_PROD (CodT1, mese, bimestre, quadrimestre, semestre, anno)

PRODUZIONE (CodC, CodT1, n_pezzi, costo_totale, costo_trasporto)

Cardinalità:

10x12 mesi X 1000 componenti = ~ 120k tuple

(in realtà gli eventi sono più sparsi)

VENDITA

TEMPO_VEN (CodT2, mese, bimestre, trimestre, anno)

CONFIGURAZIONE (CodCO, configurazione, cod_comp_scheda_video, ditta_scheda_video, cod_comp_CPU, ditta_CPU, cod_comp_HD, ditta_HD, cod_comp_RAM, ditta_RAM)

CATENA (CodCA,catena, nome, n_punti_vendita)

VENDITA (CodT2, CodCO, CodCA, costo_produzione, prezzo_vendita, n_venduti, n_difettosi)

Cardinalità:

10x12 mesi X 15 catene X 30 configurazioni = ~ 54k tuple

(in realtà gli eventi sono più sparsi)

Uniformazione delle dimensioni

La dimensione TEMPO viene condivisa da entrambi i fatti

Dimensioni:

TEMPO (CodT, mese, bimestre, trimestre, quadrimestre, semestre, anno)

COMPONENTE (CodC, cod_componente, flag_componente, tipo, capacità, velocità, dimensioni, risoluzione, ditta, regione, zona)

CONFIGURAZIONE (CodCO, configurazione, scheda_video, ditta_scheda_video, CPU, ditta_CPU, HD, ditta_HD, RAM, ditta_RAM)

CATENA (CodCA,catena, nome, n_punti_vendita)

Fatti:

PRODUZIONE (CodC, CodT, n_pezzi, costo_totale, costo_trasporto)

VENDITA (CodT, CodCO, CodCA, costo_produzione, prezzo_vendita, n_venduti, n_difettosi)

Query

Considerando solo le ditte dalle quali nel 2003 sono stati acquistati più di 100000 pezzi in totale, visualizzare per ogni ditta e tipo di componente (RAM, CPU, HD, schede video) il numero di pezzi acquistati e il costo totale dei componenti nel secondo bimestre del 2003.

```
SELECT ditta, flag_componente, SUM(n_pezzi), SUM(costo_totale)
FROM COMPONENTE C, PRODUZIONE P, TEMPO T
WHERE P.CodC=C.CodC AND P.CodT=T.CodT
AND T.bimestre="II 2003"
AND ditta IN
(
SELECT DITTA
FROM COMPONENTE C, PRODUZIONE P, TEMPO T
WHERE P.CodC=C.CodC AND P.CodT=T.CodT
AND T.anno=2003
GROUP BY DITTA
HAVING SUM(n_pezzi)>100000
)
GROUP BY ditta, flag_componente
```

Nell'anno 2007, per le memorie RAM, visualizzare per ogni quadrimestre la spesa totale sostenuta (costo componenti + costi di trasporto) e la spesa cumulativa dall'inizio dell'anno separatamente per ogni zona d'Italia e tipo di memoria RAM .

```
SELECT quadrimestre,zona,tipo, SUM(costo_totale)+SUM(costo_trasporto),
      SUM(SUM(costo_totale)+SUM(costo_trasporto)) OVER (partition BY zona,tipo ORDER
      BY quadrimestre ROWS UNBOUNDED PRECEDING)
FROM COMPONENTE C, PRODUZIONE P, TEMPO T
WHERE P.CodC=C.CodC AND P.CodT=T.CodT
AND T.anno=2007
AND flag_componente="RAM"
GROUP BY quadirmestre,zona,tipo
```

Relativamente all'anno 2002, per ogni configurazione in cui è presente la RAM acquistata dalla ditta "IntelligenceDevice", visualizzare la percentuale di prodotti difettosi rispetto al totale venduto.

```
SELECT configurazione, SUM(n_difettosi)/SUM(n_prodotti) * 100
FROM CONFIGURAZIONE C, VENDITA V, TEMPO T
WHERE V.CodCO=C.CodCO AND V.CodT=T.CodT
AND anno=2002
AND ditta_RAM="IntelligenceDevice"
GROUP by configurazione
```

VISTE

Cardinalità fatto PRODUZIONE: 10x12 mesi X 1000 componenti = ~ 120k tuple

Cardinalità fatto VENDITA: 10x12 mesi X 15 catene X 30 configurazioni = ~ 54k tuple

A – Per ogni quadrimestre del 2005 e del 2006 visualizzare il numero di CPU comprate separatamente per tipo di CPU dalla ditte del nord Italia

B – Per ogni anno visualizzare la configurazione con il miglior rapporto “costo di produzione”/”prezzo vendita”. Considerare solo le configurazioni in cui tutti la RAM e la CPU sono state prodotte dalla stessa ditta.

C – Visualizzare per ogni componente il numero di pezzi acquistati a marzo 2008 e il numero totale di pezzi acquistati nel 2008

D - Relativamente all’anno 2002, per ogni configurazione in cui è presente la RAM acquistata dalla ditta “IntelligenceDevice”, visualizzare la percentuale di prodotti difettosi rispetto al totale venduto.

E- Nell’anno 2007, per le memorie RAM, visualizzare per ogni quadrimestre la spesa totale sostenuta (costo componenti + costi di trasporto) e la spesa cumulativa dall’inizio dell’anno separatamente per ogni zona d’Italia e tipo di memoria RAM .

F – Separatamente per ogni catena di distribuzione e per ogni mese del 2008, visualizzare il numero di configurazioni diverse che sono state vendute in quel mese a quella catena.

G – Per ogni zona d’Italia e anno, visualizzare la somma totale delle spese per il trasporto dei componenti acquistati.

H - Considerando solo le ditte dalle quali nel 2003 sono stati acquistati più di 100000 pezzi in totale, visualizzare per ogni ditta e tipo di componente (RAM, CPU, HD, schede video) il numero di pezzi acquistati e il costo totale dei componenti nel secondo bimestre del 2003.

Query	Group By	Predicati
A	Quadrimestre, zona, tipo	Anno, flag componente
B	Anno, configurazione	Ditta Ram, Ditta CPU
C	Cod componente	Mese, anno
D	configurazione	Anno, Ditta RAM
E	Quadrimestre, zona, tipo	Anno, flag componente
F	Catena, mese, configurazioni	
G	Zona, anno	
H	Ditta, flag componente	Bimestre, anno

VISTA 1 A-E-G

VISTA 2 D-B

Descrizione del problema

Si vogliono analizzare alcune delle attività relative ad una ditta di trasporti europea. Nella base di dati della società sono memorizzate le informazioni di dettaglio sui viaggi svolti e sulle riparazioni effettuate sui mezzi di trasporto. La dirigenza della ditta è interessata ad analizzare i tempi di percorrenza e i costi associati ai viaggi in funzione di alcuni parametri di interesse descritti in seguito. La dirigenza è convinta che un'attenta analisi di tali informazioni permetterà all'ufficio logistica di migliorare la pianificazione dei trasporti, con un notevole risparmio economico per la ditta stessa. La dirigenza è inoltre interessata ad analizzare con attenzione le riparazioni effettuate e i costi di riparazione per i mezzi utilizzati. Tale analisi è ritenuta utile per pianificare gli acquisti di nuovi mezzi di trasporto e per decidere quali mezzi vendere o demolire tra quelli attualmente a disposizione. In merito ai viaggi la dirigenza della ditta è interessata ad analizzare la durata media (in minuti) per viaggio e il costo medio per viaggio in funzione:

- delle caratteristiche del luogo di partenza del viaggio (area geografica all'interno dello stato, stato, appartenenza alla comunità europea (si/no))
- delle caratteristiche del luogo di destinazione del viaggio (area geografica all'interno dello stato, stato, appartenenza alla comunità europea (si/no))
- tipologia del mezzo di trasporto utilizzato (furgone, autocarro, tir, ..)
- del mese, del trimestre e dall'anno in cui il viaggio è stato svolto

Relativamente alle riparazioni la ditta vuole poter analizzare il numero medio di riparazioni mensili effettuate e il costo medio per riparazione in funzione:

- del modello del mezzo di trasporto (Ducato, ...)
- della tipologia del mezzo di trasporto (furgone, autocarro, tir, ..)
- della casa produttrice del mezzo di trasporto
- dell'anno di immatricolazione del mezzo di trasporto
- del mese, del trimestre e dall'anno in cui è stata effettuata la riparazione

Sono di seguito riportate alcune delle interrogazioni di interesse per la dirigenza della ditta di trasporti:

- a) Relativamente all'anno 2003, calcolare per ogni tripletta (Area Geografica di Partenza, Area Geografica di Arrivo, Tipologia Mezzo di trasporto) la durata media per viaggio e il costo medio per viaggio.
- b) Relativamente al primo trimestre del 2005, calcolare per ogni modello di mezzo di trasporto il costo medio per riparazione. Associare ad ogni modello di mezzo di trasporto un attributo di rank legato al costo medio per riparazione (l'attributo di rank assume il valore 1 per il modello con il costo medio per riparazione più elevato).
- c) Per ogni coppia (Casa produttrice del mezzo di trasporto, Anno di Immatricolazione) calcolare il costo medio per riparazione e il numero medio di riparazioni mensili effettuate nel 2007. Ordinare i risultati in base al numero medio di riparazioni mensili effettuate.
- d) Relativamente all'anno 2004, calcolare per ogni tripletta (Area Geografica di Partenza, Area Geografica di Arrivo, Tipologia Mezzo di trasporto) la durata media per viaggio.
- e) Per ogni Casa produttrice del mezzo di trasporto calcolare il costo medio per riparazione effettuate nel 2007.

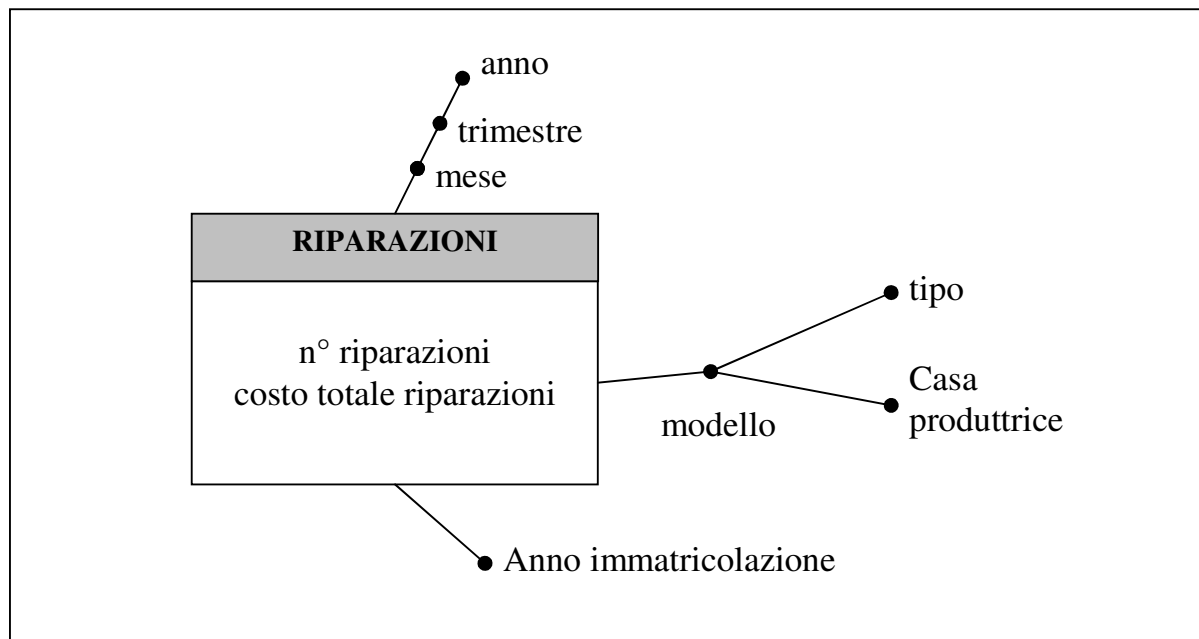
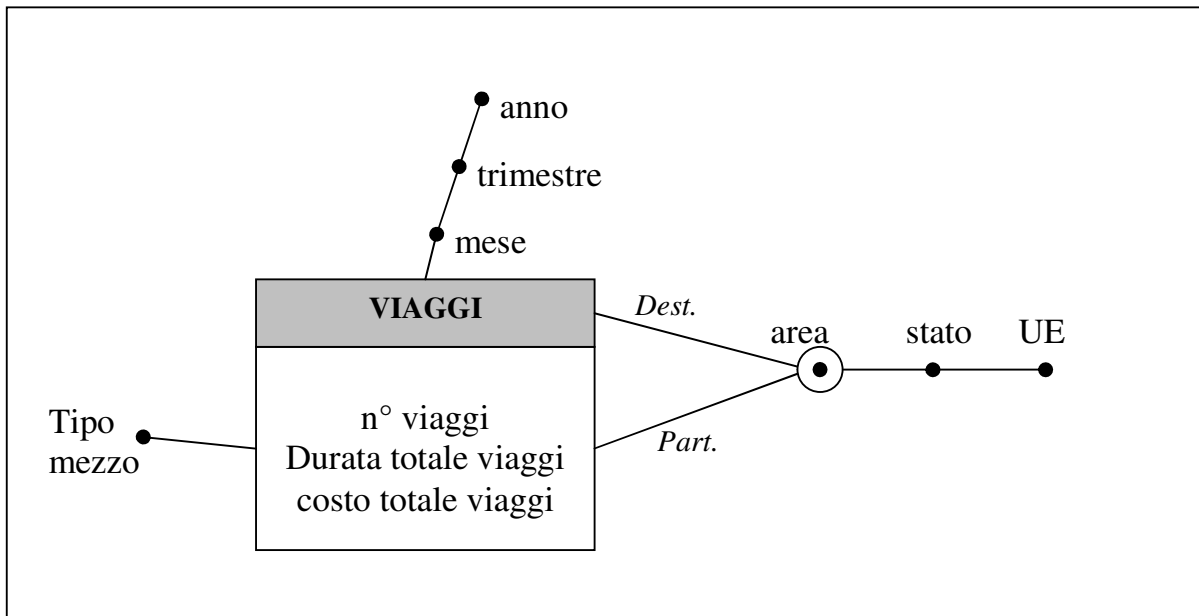
Il data warehouse realizzato deve contenere le informazioni relative agli ultimi 3 anni. Al fine di una corretta progettazione del data warehouse sono state fornite le seguenti informazioni:

- Numero di mezzi di trasporto della ditta: ~50000
- Numero di tipologie diverse di mezzi di trasporto: ~10
- Numero di modelli diversi di mezzi di trasporto: ~50
- I mezzi di trasporto più vecchi sono stati immatricolati 10 anni fa
- Numero di case produttrici di mezzi di trasporto: ~10
- Numero di aree geografiche: ~200

Progettazione

- 1.1. Progettare lo schema del data warehouse per gestire le informazioni relative ai viaggi e alle riparazioni in modo da soddisfare le richieste descritte nella descrizione del problema.
- 1.2. Decidere come gestire la dinamicità (variazioni) dei dati all'interno delle dimensioni.
- 1.3. Rispondere alle interrogazioni a), b), c).
- 1.4. Considerando le caratteristiche del data warehouse realizzato e le cardinalità delle tabelle del data warehouse progettato, decidere se e quali viste materializzate o indici potrebbe essere utile definire al fine di ottimizzare i tempi di risposta delle interrogazioni proposte nella descrizione del problema (considerare tutte le interrogazioni proposte e non solo quelle risolte in SQL al punto 1.3). Motivare le scelte fatte basandosi sulle cardinalità delle tabelle del data warehouse progettato.

Progettazione concettuale



Progettazione logica

Dimensioni:

AreeGeografiche(CodAreaGeo, AreaGeografica, Stato, AppartieneUE)

TipologieMezzi (CodTipologiaMezzo, TipologiaMezzo)

ModelliMezzi (CodModelloMezzo, ModelloMezzo, TipologiaMezzo, CasaProduttrice)

Tempo(CodTempo, Mese, Trimestre, Anno)

Fatti:

Viaggi(CodAreaGeoPart, CodAreaGeoDest, CodTipologiaMezzo, CodTempo, TotaleDurataViaggi, TotaleCostoViaggi, NumViaggi)

Riparazioni(AnnoImmatricolazione, CodModelloMezzo, CodTempo, NumeroRiparazioni, TotaleCostoRiparazioni)

Non serve un campo NumMedioRiparazioniMensili perché ogni riga corrisponde ad un mese e quindi $\text{NumMedioRiparazioniMensili} = \text{NumeroRiparazioni}$.

Una alternativa è aggiungere l'ID_mezzo alla gerarchia dei mezzi (per attaccarci direttamente la data di immatricolazione) ma questo aumenterebbe molto la cardinalità della tabella quindi è molto meglio non farlo e continuare a considerare l'“anno di immatricolazione” una dimensione a parte.

Gestione dinamicità delle dimensioni

L'unica dimensione sulla quale posso avere delle variazioni è quella relativa alle aree geografiche.

Queste possono passare da “non appartenente alla comunità europea” ad “appartenente alla comunità europea”. Ciò succedere raramente, per poche tuple e può cambiare una volta sola nella vita.

Sovrascrivo semplicemente il valore del campo AppartieneUE. (Tipo 1 - “oggi per ieri”)

Le aree geografiche le considero stabili. Se per caso ci sono dei cambiamenti ne definisco una o più nuove che vanno ad aggiungersi alle precedenti. Quelle precedenti (se non più valide) non vengono più usate. (Tipo 2)

Il Tipo 3 non è utilizzabile perché un insieme di aree può essere sostituita da una sola (devo associare tutto a quella nuova). In questo caso non mi basta sovrascrivere dei valori. Devo fondere delle tuple che facevano riferimento ad aree diverse in una sola.

Posso dover anche dividere un'area preesistente in più sottoaree. In questo caso non ho modo di sapere quanto avevo venduto in passato in ognuna delle nuove aree (sono fuse nella macro area precedente).

Interrogazioni

- a) `SELECT Part.AreaGeografica, Dest.AreaGeografica, TM.TipologiaMezzo
SUM(TotaleDurataViaggi)/SUM(NumViaggi) as ValoreDurataMediaPerViaggio,
SUM(TotaleCostoViaggi)/SUM(NumViaggi) as ValoreCostoMedioPerViaggio,
FROM AreeGeografiche Part, AreeGeografiche Dest, Tempo T, TipologieMezzi TM,
Viaggi V
WHERE V.CodAreaGeoPart=Part.CodAreaGeo
AND V.CodAreaGeoDest=Dest.CodAreaGeo
AND V.CodTempo=T.CodTempo
AND V.CodTipologiaMezzo=TM.CodTipologiaMezzo
AND T.Anno=2003
GROUP BY Part.AreaGeografica, Dest.AreaGeografica, TM.TipologiaMezzo;`
- b) `SELECT M.ModelloMezzo,
SUM(TotaleCostoRiparazioni)/SUM(NumeroRiparazioni) as
CostoMedioRiparazione,
RANK() over (ORDER BY
SUM(TotaleCostoRiparazioni)/SUM(NumeroRiparazioni) DESC as
RankCostoMedioRiparazione)
FROM ModelliMezzi M, Tempo T, Riparazioni R
WHERE R.CodTempo=T.CodTempo
AND R.CodModelloMezzo=M.CodModelloMezzo
AND T.Trimestre="1-2005" <- N.B.
GROUP BY M.ModelloMezzo;`
- c) `SELECT M.CasaProduttrice, R.AnnoImmatricolazione,
SUM(TotaleCostoRiparazione)/SUM(NumeroRiparazioni)
as CostoMedioPerRiparazione,
SUM(NumeroRiparazioni)/COUNT(distinct mese) as
NumMedioRiparazioniMensili
FROM ModelliMezzi M, Tempo T, Riparazioni R
WHERE R.CodTempo=T.CodTempo
AND R.CodModelloMezzo=M.CodModelloMezzo
AND T.Anno=2007
GROUP BY M.CasaProduttrice, R.AnnoImmatricolazione
ORDER BY NumMedioRiparazioniMensili`

Per avere il numero medio di riparazioni mensili, occorre sapere quanti mesi ci sono nel gruppo che sto condensando. In questo caso i dati sono riferiti al 2007 e si possono fare alcune considerazioni:

- posso dividere direttamente per il numero dei mesi in un anno (12) se so che la ditta ha sempre lavorato tutto l'anno
- posso fare COUNT(distinct mese). Nel caso in cui in un mese non ci sia stata nessuna riparazione (e quindi il mese non compare in nessun record del db), devo ulteriormente valutare 2 casi:
 - o in quel mese la ditta era chiusa e quindi è giusto che il COUNT non consideri quel mese
 - o in quel mese la ditta era aperta ma per motivi casuali non sono state effettuate riparazioni; in questo caso il COUNT mi restituisce un valore diverso da quello che dovrei considerare. In questo caso posso anche

ipotizzare di aggiungere nel db dei dei record con n_riparazioni=0 relativi ai mesi in cui non ho effettuato riparazioni, in modo da non alterare la somma delle riparazioni ma di far comparire cmq esplicitamente il mese, in modo che sia considerato nel COUNT

VISTE

cardinalità VIAGGI

$200 \text{ (aree partenza)} * 200 \text{ (aree arrivo)} * 3 \times 12 \text{ (anni x mesi)} * 10 \text{ (tipo dei mezzi di trasporto)} = 14.4 \text{ M}$

Cardinalità RIPARAZIONI

$36 \text{ (mesi x anni)} * 50 \text{ (modelli)} * 10 \text{ (anni di immatricolazione)} = 18\text{k}$

query	Group by	predicati
A	Area_part, area_arr, tipo_mezzo	Anno
B	Modello_mezzo	trimestre
C	Casa_prod, anno_immatr	Anno
D	Area_part, area_arr, tipo_mezzo	anno
e	Casa_prod	Anno

Query a) $200 * 200 * 10 = 400\text{k}$ (ben al di sotto del numero di tuple del fatto Viaggi quindi conviene generare una vista)

Area_partenza X Area_arrivo X tipo_mezzo X Anno

Query b) 50 (conviene creare una vista)

Query c) $10 * 10 = 100$ (conviene creare una vista)

Casa_prod X Anno_imm X Anno

Query d) è molto simile alla query a). a questo punto genero una sola vista per la query a) e d) per rispondere a entrambe le richieste (800k record)

Query e) posso rispondere con la stessa vista della query c)

Descrizione del problema

Un network televisivo gestisce diversi canali tematici e vorrebbe analizzare i dati sugli ascolti nei programmi per ottimizzare il palinsesto e i profitti della pubblicità. I canali sono caratterizzati da un nome ("CartoonNetwork", "TuttoTennis", "Natura&Co.",...) e trattano un tema specifico (film, sport, politica, economia,...).

Il network vorrebbe, incrociando i dati del palinsesto dei suoi canali con i dati forniti dall'Auditel, capire quali sono i momenti della giornata in cui è più proficuo inserire le pubblicità pagate dagli sponsor.

L'Auditel fornisce ad intervalli di 15 minuti, per ogni canale il numero massimo e minimo di ascoltatori sintonizzati. In più, per ogni singolo programma tv, l'Auditel fornisce lo share (percentuale di spettatori sul totale) del programma e una statistica sulla provenienza geografica (provincia) degli ascoltatori.

La società è interessata ad analizzare gli andamenti del numero massimo e minimo di spettatori e i minuti di pubblicità trasmessi in funzione di:

- l'ora del giorno, il giorno (e se è festivo o no), il giorno della settimana e il mese, anno
- il quarto d'ora del giorno e la mezz'ora del giorno
- il periodo della giornata: notte (dalle 0 alle 6:30), mattino (6:30-12:30), pomeriggio (12:30-19:30), sera (19:30, 0)
- della fascia oraria consigliata: adulti (dalle 0 alle 7), bambini (7-15), tutti (15-0)
- del canale televisivo e del tema del canale

In più per ogni programma tv si è interessati ad analizzare lo share ottenuto, i minuti totali di pubblicità trasmessi e il numero di spot trasmessi in funzione di:

- nome del programma e il canale che lo trasmette
- tipo di programma (notiziario, talk show, evento, telefilm,...)
- giorno e mese
- provincia e regione degli ascoltatori

Il data warehouse realizzato deve contenere le informazioni relative agli ultimi 5 anni. Al fine di una corretta realizzazione del data warehouse sono state fornite le seguenti informazioni:

- Numero di canali: ~40
- Numero di programmi: ~500
- Numero di tipi di programmi: ~50
- Numero di temi: ~15
- Numero di province: ~100
- Numero di regioni: ~20

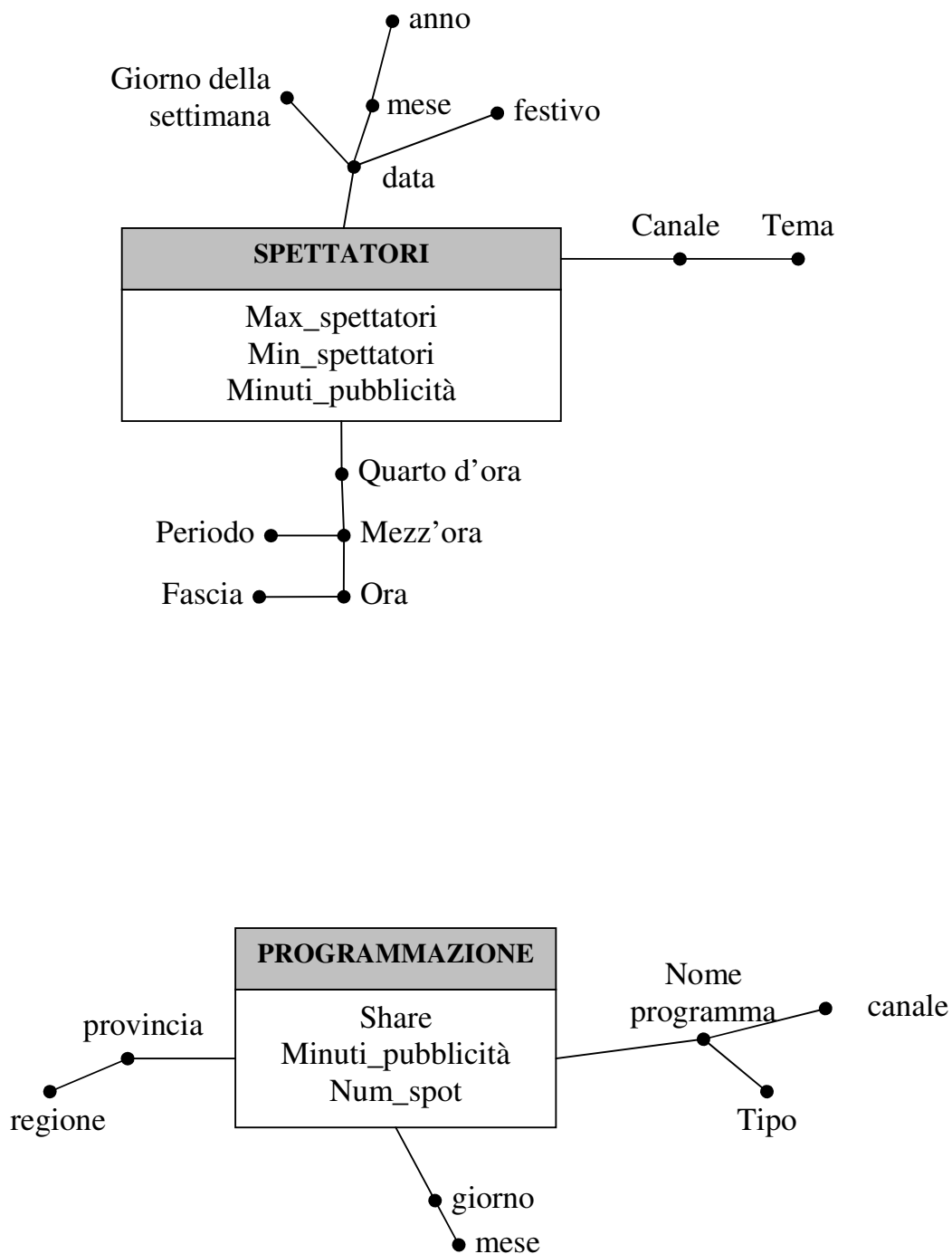
Gli analisti vogliono poter disporre delle seguenti informazioni:

- 1) Relativamente alla regione Piemonte, considerando solo i programmi di tipo "notiziario" con una media giornaliera di almeno il 10% di share nel mese di Gennaio 2008, trovare la durata media degli spot trasmessi.
- 2) Trovare la media del numero massimo di spettatori sintonizzati sui canali sportivi tra le 12 e le 13 dei giorni festivi.
- 3) Nel 2007 trovare il totale dei minuti di pubblicità trasmessa di lunedì in ogni canale.
- 4) Trovare il "periodo della giornata" con il minor numero di spettatori del canale "TuttoTennis".
- 5) Relativamente ai canali tematici "sportivi" nel mese di gennaio 2008, trovare la classifica delle ore del giorno con più minuti di pubblicità (dall'ora del giorno con più pubblicità a quella con meno).
- 6) Nel 2007, trovare per ogni canale il massimo numero di spettatori sintonizzati contemporaneamente.
- 7) Trovare il programma nel quale il 1 Gennaio 2008 sono stati trasmessi più spot.

Progettazione

- Progettare lo schema concettuale e logico relazionale del data warehouse necessario per gestire le informazioni richieste dalla dirigenza, valutando anche eventuali uniformazioni di dimensioni.
- Decidere come gestire la dinamicità (variazione) dei dati all'interno delle dimensioni.
- Scrivere le interrogazioni in SQL corrispondenti alle interrogazioni frequenti 1), 5) e 6).
- Considerando le caratteristiche del data warehouse realizzato e la cardinalità dei dati memorizzati nel data warehouse, decidere se e quali viste materializzate potrebbe essere utile definire al fine di ottimizzare i tempi di risposta delle interrogazioni proposte nelle specifiche del problema (considerare **tutte** le interrogazioni proposte e non solo quelle risolte in SQL. Motivare le scelte fatte.

Progettazione concettuale



Progettazione logica

- DISPONIBILITA

ORA (CodO, Quarto_ora, mezz_ora, ora, periodo, fascia)

DATA1 (CodD, Giorno, giorno_settimana, festivo, mese, anno)

CANALE (CodC, nome_canale, tema)

LUOGO (CodL, zona_città, città, provincia, regione, zona_italia)

SPETTATORI (CodO, CodD, CodC, max_spettatori, min_spettatori, minuti_pubblicità)

- PROGRAMMAZIONE

PROGRAMMA (CodP, nome_programma, nome_canale, tipo_canale)

DATA2 (CodD, giorno, mese)

LUOGO (CodL, provincia, regione)

DISPONIBILITA (CodP, CodD, CodL, share, minuti_pubblicità, num_spot)

Si possono condividere alcune dimensioni. Lo schema logico risultante è quindi il seguente:

- Dimensioni condivise
 - DATA1 e DATA2

Query

- Relativamente alla regione Piemonte, considerando solo i programmi di tipo “notiziario” con una media giornaliera di almeno il 10% di share nel mese di Gennaio 2008, trovare la durata media degli spot trasmessi

```
SELECT SUM(minuti_pubblicità)/SUM(numero_spot)
FROM programmazione, luogo, tempo
WHERE {.. join..}
AND regione="piemonte"
AND mese="Gennaio 2008"
AND ID_programma IN
(
SELECT ID_programma
FROM programmazione, luogo, programmi, tempo
WHERE {.. join..}
AND regione="piemonte"
AND mese="Gennaio 2008"
AND tipo_programma="notiziario"
GROUP BY ID_programma
HAVING (SUM(share)/COUNT(distinct data))>10
)
```

- Relativamente ai canali tematici “sportivi” nel mese di gennaio 2008, trovare la classifica delle ore del giorno con più minuti di pubblicità (dall’ora del giorno con più pubblicità a quella con meno)

```
SELECT ora_del_giorno, RANK() OVER (ORDER BY SUM(minuti_pubblicità) DESC) as
classifica
FROM Spettatori, tempo, canale
WHERE {.. join..}
AND mese="Gennaio 2008"
AND tema_canale="sportivo"
GROUP BY ora_del_giorno
ORDER BY classifica DESC
```

- Nel 2007, trovare per ogni canale il massimo numero di spettatori sintonizzati contemporaneamente

```
SELECT Canale, MAX(max_spettatori)
FROM Spettatori, tempo, canale
WHERE {.. join..}
AND anno=2007
GROUP BY Canale
```

Viste

Cardinalità fatto SPETTATORI = $40 * (365*5) * (24*4) = 7 \text{ M circa}$

Cardinalità fatto PROGRAMMAZIONE = $500 * 100 * (365*5) = 91 \text{ M circa}$

	Misure	Dimensioni	Predicati
1	Share, min_pubblicità, num_spot		regione, mese, tipo_programma, data, programma
2	Max_spettatori		Tema_canale, ora, festivo
3	Min_pubblicità	canale	Anno, giorno_settimana
4	Min_spettatori	Periodo_giornata	Canale
5	Min_pubblicità	ora	Tema_canale, mese
6	Max_spettatori	canale	Anno
7	Num_spot	programma	data

	Tempo	Programma	Luogo	Ora
1	Data	Programma	regione	
2	Festivo	Tema		Ora
3	Anno, giorno_settimana	Canale		
4		Canale		Periodo
5	Mese	Tema		ora
6	Anno	Canale		
7	Data	programma		

VISTE:

Per (data-programma) risponde alle query 3-6-7

Cardinalità $(365*5)*500 = 900\text{k record circa}$

Per (data-ora-tema) risponde alle query 2-5

Cardinalità $(365*5)*15 = 27\text{k record circa}$