on views

what are views usefull for?

outline

1. what is a view?
2. processing queries involving views
3. what is a materialized view?
4. processing queries using materialized views
5. recursive views (see e.g., notes of CS145 from Stanford U.)

views

Movies[title,year,length,genre,studioName,producer]
Producer[name,nationality]

```
CREATE VIEW MovieProd(movieTitle,prodNat) AS
SELECT title,nationality
FROM Movies, Producer
WHERE producer = name
```

put differently
MovieProd(t,n) ← Movies(t,y,g,s,p),Producer(p,n)

querying views

- a view is not a table
- it is an intensional definition
- its extension is not stored
- but it may be queried as a table

```
SELECT starName,ProdNat FROM MovieProd, StarsIn WHERE
title=movieTitle
```

processing queries using views

SELECT V.x FROM V WHERE φ (i.e., $\pi_{V.x}(\sigma_{\varphi}(V)))$

suppose V is defined by : $\pi_{y,x}(\sigma_{\theta}(R))$

processing Q means processing : $\pi_{x}(\sigma_{\varphi}(\pi_{y,x}(\sigma_{\theta}(R))))$

and thus: $\pi_{x}(\sigma_{\varphi\wedge\theta}(R)))$

materialized views

```
CREATE MATERIALIZED VIEW MovieProd(movieTitle,prodNat)
AS
SELECT title,nationality
FROM Movies, Producer
WHERE producer = name
```

materialized views

- a materialized view is not a table
- even though its extension *is* stored
- it *remains* an intensional definition
- and it may be queried as a table

if tuples are INSERTed (resp. DELETEd) in (resp. FROM) e.g., Movies, then MovieProd has to be refreshed

views maintenance

for simple views, maintenance can be incremental

- consider MovieProd defined as
  MovieProd(t,n) ← Movies(t,y,g,s,p),Producer(p,n)
- suppose (a,b,c,d,e) is inserted into Movies
- what to insert into MovieProd?
- we have the title: $\pi_{title}(a,b,c,d,e)$
- SELECT nationality FROM Producer WHERE name='e'

views maintenance

for simple views, maintenance can be incremental

- consider MovieProd defined as MovieProd(t,n) ← Movies(t,y,g,s,p),Producer(p,n)
- suppose (a,b) is inserted into Producer
- what to insert into MovieProd?
- INSERT INTO MovieProd SELECT title,'b' FROM Movies,Producer WHERE name='a'

views maintenance

for simple views, maintenance can be incremental

- consider MovieProd defined as
  MovieProd(t,n) ← Movies(t,y,g,s,p),Producer(p,n)
- suppose (a,b,c,d,e) is deleted from Movies
- what to delete from MovieProd?
- DELETE FROM MovieProd WHERE title='a'

what if we DELETE (a,b) FROM Producer?

rewriting queries to use materialized views

assume a materialized view v defined by

SELECT L_v FROM R_v WHERE C_v

- L_v is a list of attributes
- R_v is a list of relations
- C_v is a condition

rewriting queries to use materialized views

suppose a query q defined by

SELECT L_q FROM R_q WHERE C_q

- L_q is a list of attributes
- R_q is a list of relations
- C_q is a condition

can we rewrite q to use what v materializes?

conditions for rewriting

if all of the following apply

1. $R_v \subseteq R_q$
2. $C_q \Rightarrow C_v$
3. if $C_q \equiv C_v \wedge C$ then the attributes of relations on R_v that C mentions are attributes on L_v
4. the attributes on L_q that come from R_v are also on L_v

rewriting

if the conditions expressed above are met, then

1. replace R_q by V and the relations in $R_q \setminus R_v$
2. replace C_q by C

example

assume the following view is materialized

```
CREATE MATERIALIZED VIEW MovieProd(movieTitle,prodNat)
AS
SELECT title,nationality FROM Movies, Producer
WHERE producer = name
```

then the query

```
SELECT starName,ProdNat FROM Movies, Producer, StarsIn
WHERE title=movieTitle AND producer=name
```

can be rewritten

```
SELECT starName,ProdNat FROM MovieProd, StarsIn WHERE
title=movieTitle
```

example

assume the following view is materialized

MovieProd(t,n) ← Movies(t,y,g,st,p),Producer(p,n)

then the query

ans(t,n) ← Movies(t,y,g,st,p),Producer(p,n),StarsIn(t,y,s)

can be rewritten

ans(s,n) ← MovieProd(t,n),StarsIn(t,y,s)