Keys

- SuperKey
  - a *set* of attributes whose values together *uniquely* identify a tuple in a relation
- Candidate Key
  - a superkey for which no proper subset is a superkey…a key that is *minimal* .
  - *Can be more than one for a relation*
- Primary Key
  - a candidate key chosen to be the main key for the relation.
  - *One for each relation*
- Keys can be *composite*

e.g.: Staff(lecturer,roomno,appraiser)

SK = {lecturer,roomno,appraiser},
{lecturer,roomno}, {lecturer, appraiser},
{roomno,appraiser}, {lecturer} and {roomno}

CK = {lecturer} and {roomno}

PK = {lecturer}

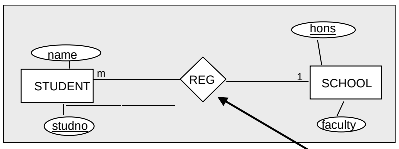
Foreign Key

- a (set of) attribute(s) in a relation that exactly matches a (primary) key in another relation
  - the names of the attributes don't have to be the same but must be of the same domain
  - a foreign key in a relation A matching a primary key in a relation B represents a
- *many:one* relationship between A and B

Student(studno,name,tutor,year)

Staff(lecturer,roomno,appraiser)

Relationship -v- Relation



Relationship

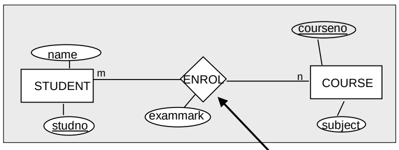
"an association between several entities represented by a *Relationship Type* of which there will be many *Relationship Instances"*

STUDENT (studno, name, hons)

SCHOOL(hons, faculty)

Relation (table)

Relationship -v- Relation



ENROL Relationship

STUDENT (studno, name)

ENROL(studno, courseno,exammark)

COURSE(courseno, subject)

ENROL Relation

Joins

STUDENT

studno	name	tutor
s1	jones	bush
s2	brown	kahn
s3	smith	goble

STAFF

lecturer	roomno
kahn	IT206
bush	2.26

e.g., get studno, name and tutor's roomno for each student

STUDENT

studno	name	tutor
s1	jones	bush
s2	brown	kahn
s3	smith	goble

STAFF

lecturer	roomno
kahn	IT206
bush	2.26

STUDENT

studno	name	tutor
s1	jones	bush
s2	brown	kahn
s3	smith	goble

STAFF

lecturer	roomno
kahn	IT206
bush	2.26

$\pi_{studno,name,roomno}$ (STUDENT $\times_{tutor=lecturer}$ STAFF)

studno	name	tutor	lecturer	roomno
s1	jones	bush	kahn	IT206
s1	jones	bush	bush	2.26
s2	brown	kahn	kahn	IT206
s2	brown	kahn	bush	2.26
s3	smith	bush	kahn	IT206
s3	smith	bush	bush	2.26

SELECT

FROM

STUDENT a, STAFF b

WHERE

STUDENT

studno	name	tutor
s1	jones	bush
s2	brown	kahn
s3	smith	goble

STAFF

lecturer	roomno
kahn	IT206
bush	2.26

$\pi_{studno,name,roomno}$ (STUDENT $\bowtie_{tutor=lecturer}$ STAFF)

studno	name	tutor	lecturer	roomno
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
s1	jones	bush	bush	2.26
s2	brown	kahn	kahn	IT206
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
s3	smith	bush	bush	2.26

SELECT

FROM

STUDENT a, STAFF b

WHERE

a.tutor=b.lecturer

STUDENT

studno	name	tutor
s1	jones	bush
s2	brown	kahn
s3	smith	goble

STAFF

lecturer	roomno
kahn	IT206
bush	2.26

$\pi_{studno,name,roomno}$ (STUDENT $\bowtie_{tutor=lecturer}$ STAFF)

studno	name	[redacted]	[redacted]	roomno
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
s1	jones	[redacted]	[redacted]	2.26
s2	brown	[redacted]	[redacted]	IT206
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
s3	smith	[redacted]	[redacted]	2.26

SELECT

a.studno, a.name, b.roomno

FROM

STUDENT a, STAFF b

WHERE

a.tutor=b.lecturer

STUDENT

studno	name	tutor
s1	jones	bush
s2	brown	kahn
s3	smith	goble

STAFF

lecturer	roomno
kahn	IT206
bush	2.26

$\sigma_{studno='s1'}(\pi_{studno,name,roomno}$ (STUDENT $\bowtie_{tutor=lecturer}$ STAFF))

studno	name	[redacted]	[redacted]	roomno
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
s1	jones	[redacted]	[redacted]	2.26
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

SELECT

a.studno, a.name, b.roomno

FROM

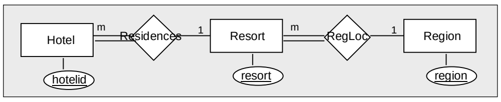
STUDENT a, STAFF b

WHERE

a.tutor=b.lecturer and

studno='s1'

Join as Path



Get hotels and their regions

SELECT

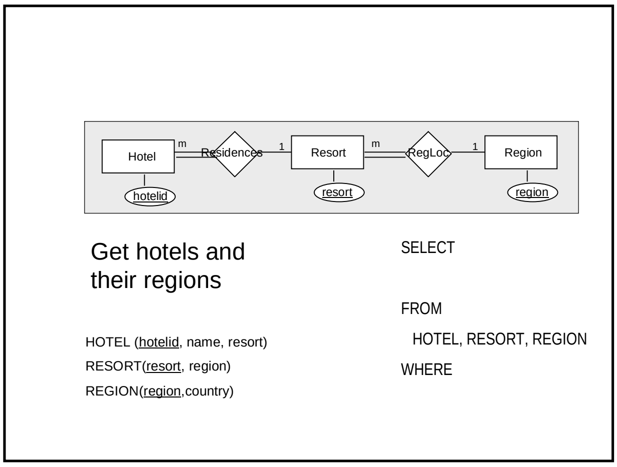
FROM

WHERE

HOTEL (hotelid, name, resort)

RESORT(resort, region)

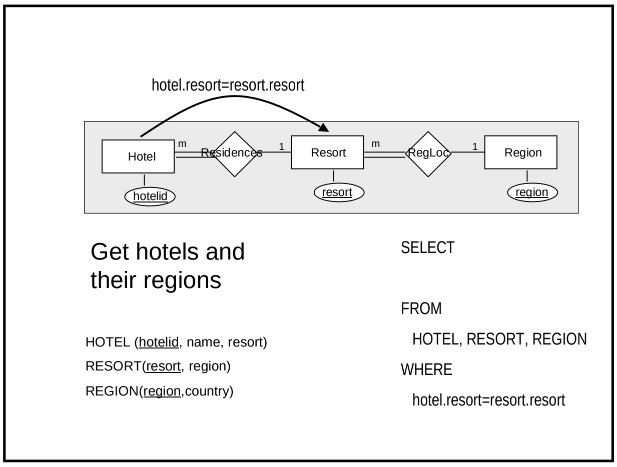
REGION(region,country)



Get hotels and their regions

HOTEL (hotelid, name, resort)
RESORT(resort, region)
REGION(region,country)

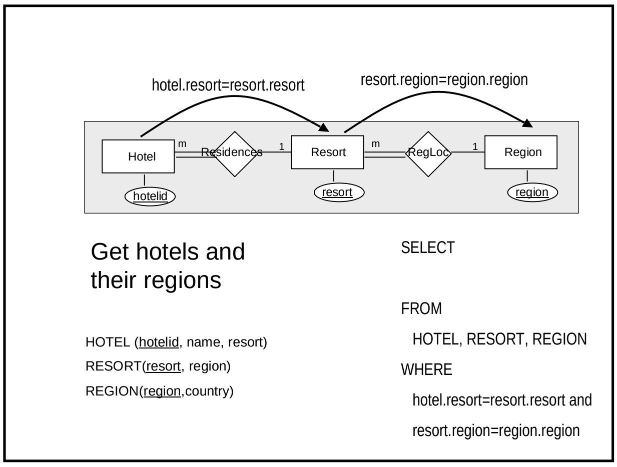
```
SELECT

FROM
  HOTEL, RESORT, REGION
WHERE
```



Get hotels and their regions

HOTEL (hotelid, name, resort)
RESORT(resort, region)
REGION(region,country)

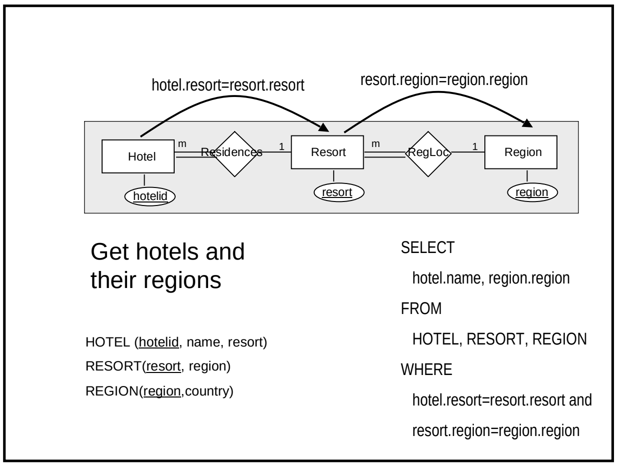
```
SELECT

FROM
  HOTEL, RESORT, REGION
WHERE
  hotel.resort=resort.resort
```



Get hotels and their regions

HOTEL (hotelid, name, resort)
RESORT(resort, region)
REGION(region,country)

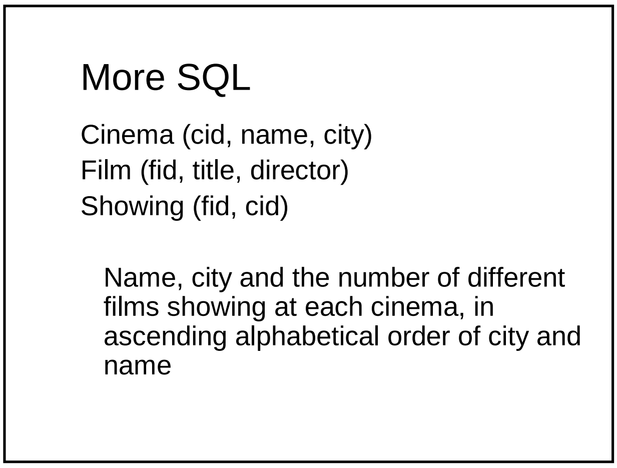
```
SELECT

FROM
  HOTEL, RESORT, REGION
WHERE
  hotel.resort=resort.resort and
  resort.region=region.region
```



Get hotels and their regions

HOTEL (hotelid, name, resort)
RESORT(resort, region)
REGION(region,country)

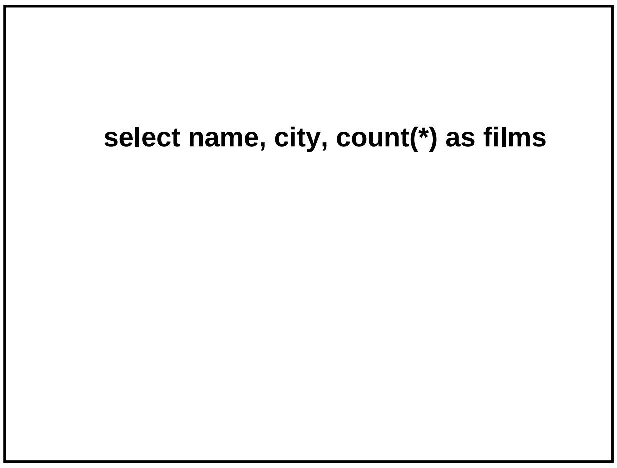
```
SELECT
  hotel.name, region.region
FROM
  HOTEL, RESORT, REGION
WHERE
  hotel.resort=resort.resort and
  resort.region=region.region
```



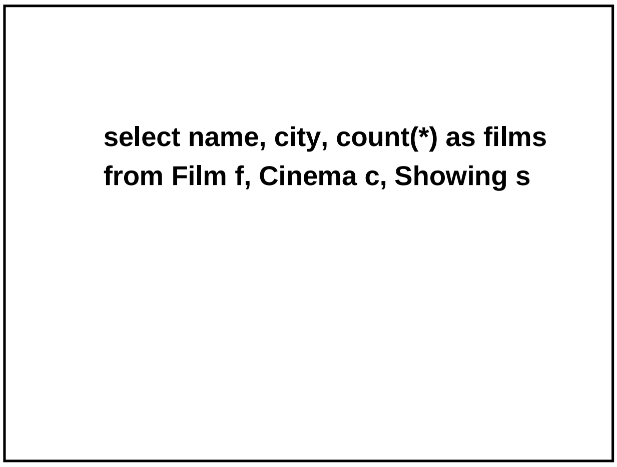
More SQL

Cinema (cid, name, city)
Film (fid, title, director)
Showing (fid, cid)

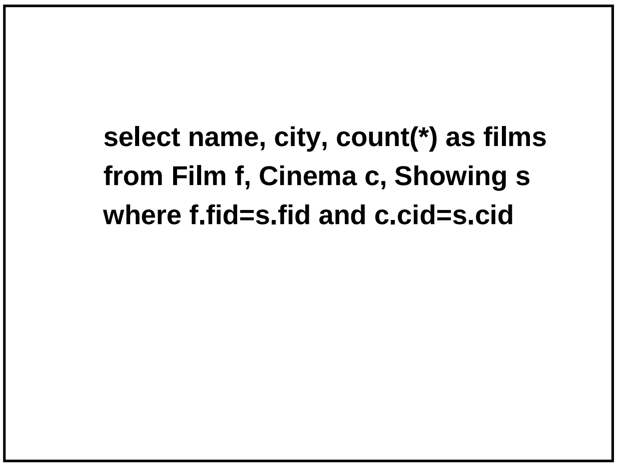
Name, city and the number of different films showing at each cinema, in ascending alphabetical order of city and name



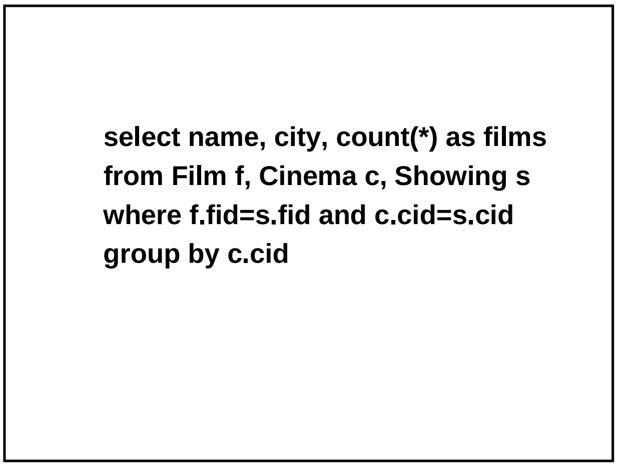
select name, city, count(*) as films



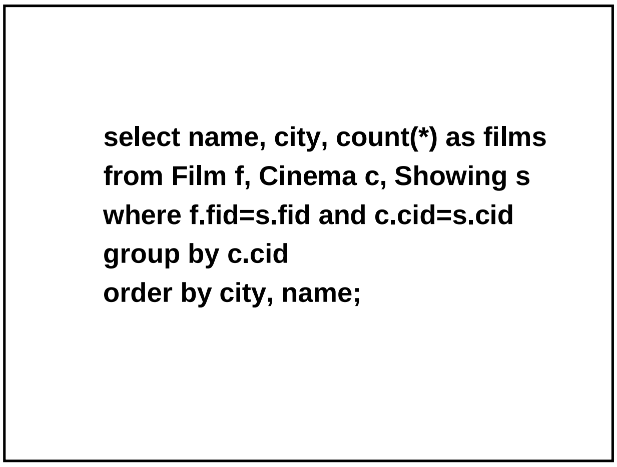
```
select name, city, count(*) as films
from Film f, Cinema c, Showing s
```



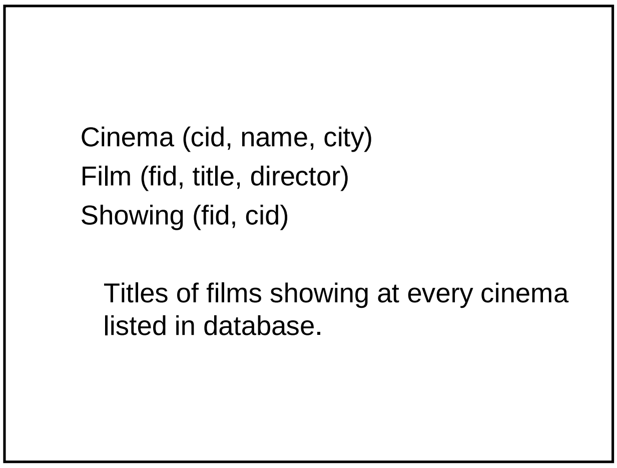
```
select name, city, count(*) as films
from Film f, Cinema c, Showing s
where f.fid=s.fid and c.cid=s.cid
```



```
select name, city, count(*) as films
from Film f, Cinema c, Showing s
where f.fid=s.fid and c.cid=s.cid
group by c.cid
```

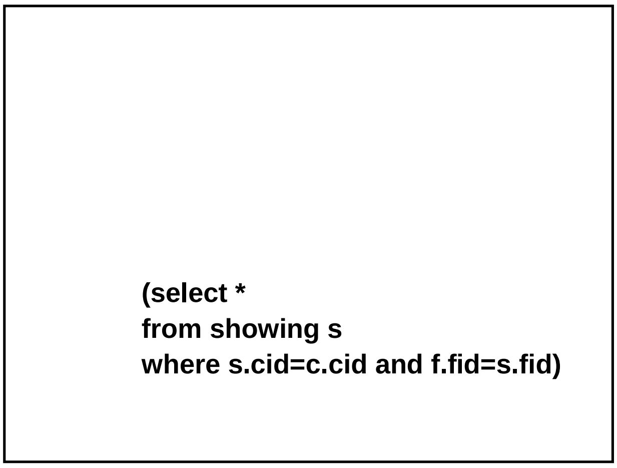


```
select name, city, count(*) as films
from Film f, Cinema c, Showing s
where f.fid=s.fid and c.cid=s.cid
group by c.cid
order by city, name;
```

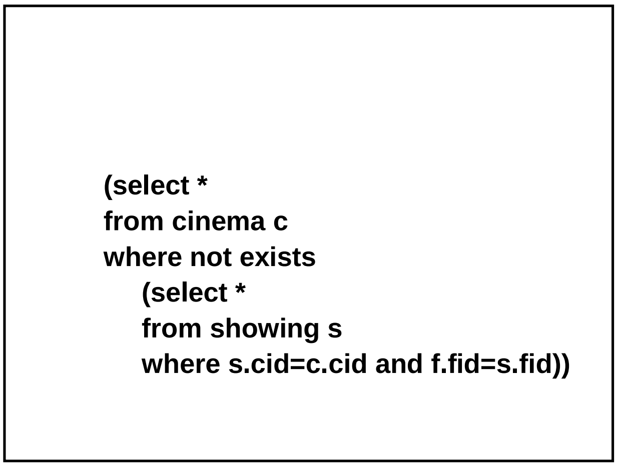


Cinema (cid, name, city)
Film (fid, title, director)
Showing (fid, cid)

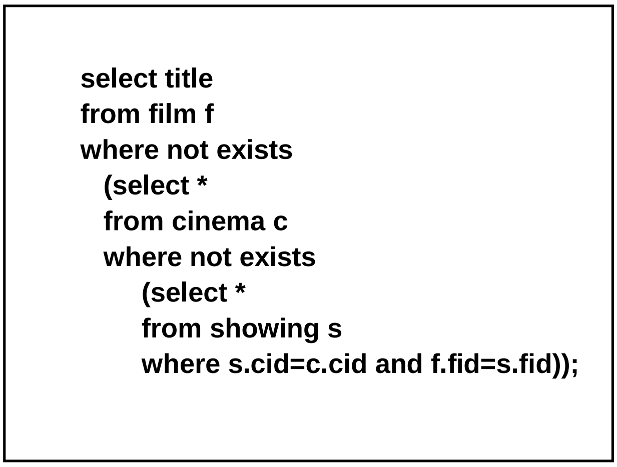
Titles of films showing at every cinema listed in database.



```
(select *
from showing s
where s.cid=c.cid and f.fid=s.fid)
```



```
(select *
from cinema c
where not exists
    (select *
    from showing s
    where s.cid=c.cid and f.fid=s.fid))
```



```
select title
from film f
where not exists
  (select *
  from cinema c
  where not exists
      (select *
      from showing s
      where s.cid=c.cid and f.fid=s.fid));
```