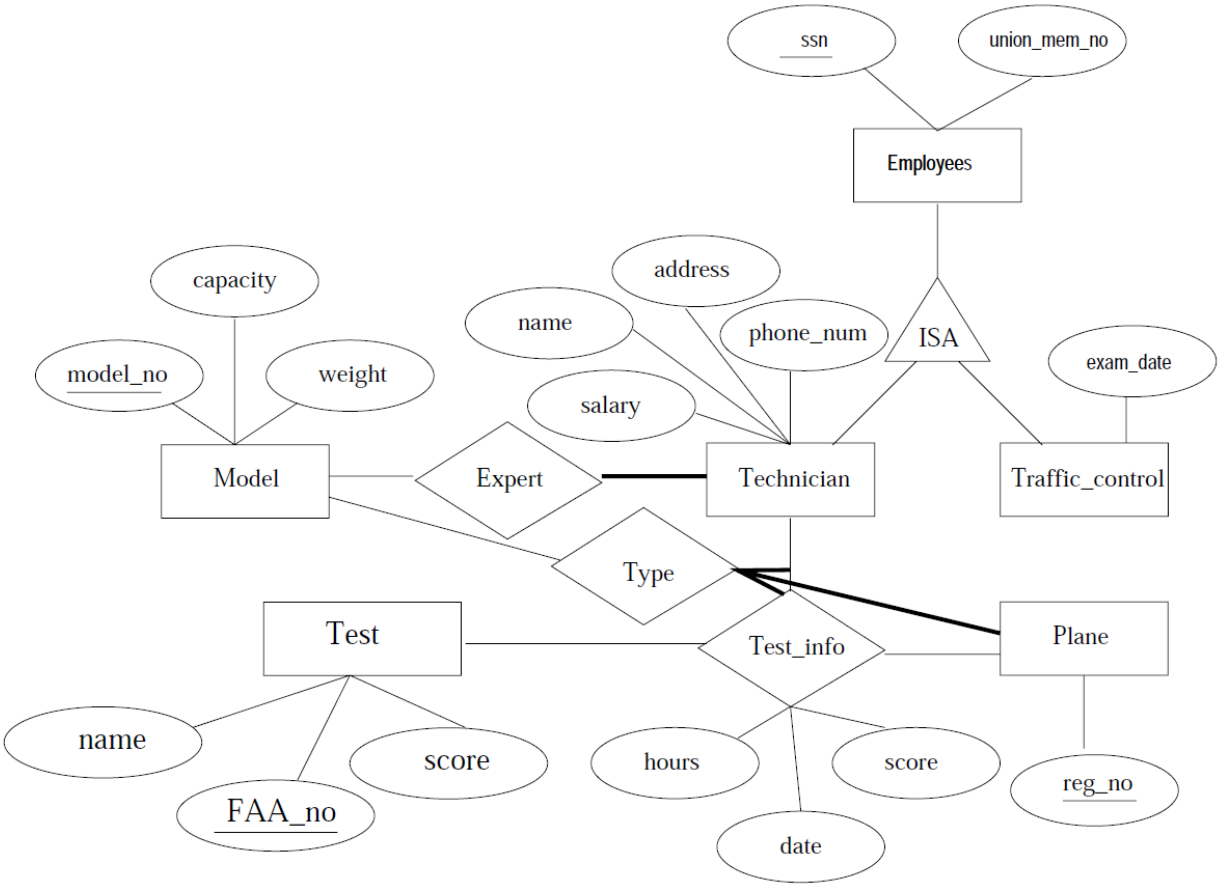
ER Diagram



1. Since all airline employees belong to a union, there is a covering constraint on the Employees ISA hierarchy.
2. You cannot note the expert technician constraint the FAA requires in an ER diagram. There is no notation for equivalence in an ER diagram and this is what is needed: the Expert relation must be equivalent to the Type relation.

Relational Model

The following SQL statements create the corresponding relations.
1. CREATE TABLE Expert (ssn CHAR(11),
model no INTEGER,
PRIMARY KEY (ssn, model no),
FOREIGN KEY (ssn) REFERENCES Technician,
FOREIGN KEY (model no) REFERENCES Models)
The participation constraint cannot be captured in the table.

2. CREATE TABLE Models (model no INTEGER,
capacity INTEGER,

weight INTEGER,
PRIMARY KEY (model no))

3. CREATE TABLE Employees (ssn CHAR(11),
union mem no INTEGER,
PRIMARY KEY (ssn))

4. CREATE TABLE Technician emp (ssn CHAR(11),
name CHAR(20),
address CHAR(20),
phone no CHAR(14),
PRIMARY KEY (ssn),
FOREIGN KEY (ssn)
REFERENCES Employees
ON DELETE CASCADE)

5. CREATE TABLE Traffic control emp (ssn CHAR(11),
exam date DATE,
PRIMARY KEY (ssn),
FOREIGN KEY (ssn)
REFERENCES Employees
ON DELETE CASCADE)

6. CREATE TABLE Plane Type (reg no INTEGER,
model no INTEGER,
PRIMARY KEY (reg no),
FOREIGN KEY (model no) REFERENCES Models)

7. CREATE TABLE Test info (FFA no INTEGER,
ssn CHAR(11),
reg no INTEGER,
hours INTEGER,
date DATE,
score INTEGER,
PRIMARY KEY (ssn, reg no, FFA no),
FOREIGN KEY (reg no) REFERENCES Plane Type,
FOREIGN KEY (FAA no) REFERENCES Test,
FOREIGN KEY (ssn) REFERENCES Employees)

8. The constraint that tests on a plane must be conducted by a technician who is an expert on that model can be expressed in SQL as follows.
CREATE TABLE Test info (FFA no INTEGER,
ssn CHAR(11),

reg no INTEGER,
hours INTEGER,
date DATE,
score INTEGER,
PRIMARY KEY (ssn, reg no, FFA no),
FOREIGN KEY (reg no) REFERENCES Plane Type,
FOREIGN KEY (FAA no) REFERENCES Test,
FOREIGN KEY (ssn) REFERENCES Technician emp)
CONSTRAINT MODEL
CHECK (SELECT * FROM Expert, Type
WHERE Expert.ssn = ssn AND
Expert.model no = Type.model no AND
Type.reg no = reg no)

Relational Algebra

1. List names of concerts halls in Chicago whose capacity exceeds 10,000.

SQL:
SELECT hname
FROM Hall H
WHERE H.cname='Chicago' AND H.capacity > 10,000

RA:
$\pi_{hname}(\sigma_{cname= \text{'Chicago'} \wedge capacity>10,000} \text{Hall})$

DRC:
$\{ <Hn> \mid \exists Cn,Ca(<Hn,Cn,Ca> \in \text{Hall} \wedge Cn=\text{'Chicago'} \wedge Ca > 10,000)\}$

2. List names of artists who performed before at least 10,000 spectators, together with cities where those performance took place.

SQL:
SELECT S.artist,H.cname
FROM Show S, Hall H
WHERE H.hname=S.hname AND S.attendance > 10,000

RA:
$\pi_{artist,cname}(\sigma_{attendance>10,000} \text{Show} \bowtie \text{Hall})$

DRC:
$\{ <Ar,Cn> \mid \exists T,Hn,At(<T,Hn,At,Ar> \in \text{Show} \wedge At > 10,000 \wedge \exists Ca(<Hn,Cn,Ca> \in \text{Hall}))\}$

3. Find all the states in USA where Springsteen has performed.

SQL:

SELECT C.state
FROM Show S, Hall H, City C
WHERE S.hname=H.hname AND H.cname=C.cname AND S.artist='Springsteen' AND C.country='USA'

RA:
$\pi_{state}(\sigma_{artist= \text{'Springsteen'} \wedge country='USA'} \text{Show} \bowtie \text{Hall} \bowtie \text{City})$

DRC:
{ <St> | $\exists$T,Hn,At(<T,Hn,At,Springsteen> $\in$ Show $\wedge \exists$Cn,Ca(<Hn,Cn,Ca> $\in$ Hall $\wedge \exists$ <Cn,USA,St> $\in$ City))}

4. Find all the artists who performed at least two times in Chicago.

SQL:
SELECT S.artist
FROM Show S, Hall H
WHERE S.hname=H.hname AND H.cname='Chicago'
GROUP BY S.artist
HAVING COUNT(*)>1

RA:
$\rho(T1, \pi_{artist,title}(\sigma_{cname='Chicago'} \text{Show} \bowtie \text{Hall}))$
$\rho(T2(1->a1,2->t1,3->a2,4->t2), T1 \square T1)$
$\pi_{a1}(\sigma_{a1=a2 \wedge t1<>t2} T2)$

DRC:
{ <Ar> | $\exists$T,Hn,At(<T,Hn,At,Ar> $\in$ Show $\wedge \exists$Ca(<Hn,Chicago,Ca> $\in$ Hall)) $\exists$T',Hn',At', Ar'(<T',Hn',At',Ar'> $\in$ Show $\wedge \exists$Ca'(<Hn',Chicago,Ca'> $\in$ Hall) $\wedge$ Ar=Ar' $\wedge$ T<>T')}

5. List shows that played in every city in Indiana, USA.

SQL:
SELECT S.title
FROM Show S
WHERE NOT EXISTS (
(SELECT C1.cname
FROM City C1
WHERE C1.country='USA' AND C1.state='Indiana')
EXCEPT
(SELECT H2.cname
FROM Show S2, Hall H2
WHERE S2.title=S.title AND S.hname=H2.hname

)

RA:
$\pi_{title,cname}$Show $\bowtie$Hall \ $\pi_{cname}\sigma_{country=''USA' \wedge state='Indiana'}$City

DRC:
{ <T> | $\exists$Hn,At,Ar(<T,Hn,At,Ar> $\in$ Show $\wedge \forall$ <Cn,USA,Indiana> $\in$ City($\exists$ <Hn,Cn,Ca> $\in$ Hall)) }

6. Find artists who performed only in Chicago or West Lafayette.

SQL:
SELECT S.artist
FROM Show S
EXCEPT
SELECT S.artist
FROM Show S, Hall H
WHERE S.hname=H.hname AND H.cname<>'Chicago' AND H.cname<>'West Lafayette')

RA:
π_{artist}Show - π_{artist}($\sigma_{cname\neq'Chicago' \wedge cname\neq'West\ Lafayette'}$ Show $\bowtie$Hall)

DRC:
{ <Ar> | $\exists$T,Hn,At(<T,Hn,At,Ar> $\in$ Show $\wedge \neg\exists$ Cn,Ca(<Hn,Cn,Ca> $\in$ Hall $\wedge$ Cn $\neq$ 'Chicago' $\wedge$ Cn $\neq$ 'West Lafayette')) }

7. Find artists all of whose shows are sold out.

SQL:
SELECT S.artist
FROM Show S, Hall H
WHERE S.hname=H.hname
GROUP BY S.artist
HAVING EVERY(S.attendance=H.capacity)

RA:
$\pi_{artist,title}$Show \ π_{title}($\sigma_{attendace=capacity}$ Show $\bowtie$Hall)

DRC:
{ <Ar> | $\exists$T,Hn,At(<T,Hn,At,Ar> $\in$ Show $\wedge \forall$ <Hn,Cn,Ca> $\in$ Hall(At=Ca)) }