Advanced Database Management Systems

Lecture 5 – Chapter 7
ER to Relational Schema Translation

NEXT UP

- skip ahead to Chapter 7:

  Translating ER Schemas to Relational Schemas

- then back to Chapter 6:

  The Relational Algebra: operations on relations

REVIEW

- Specify all ***foreign keys*** for the following schema:

STUDENT(SID, Name, Major, BirthDate)

COURSE(CID, Name, Dept)

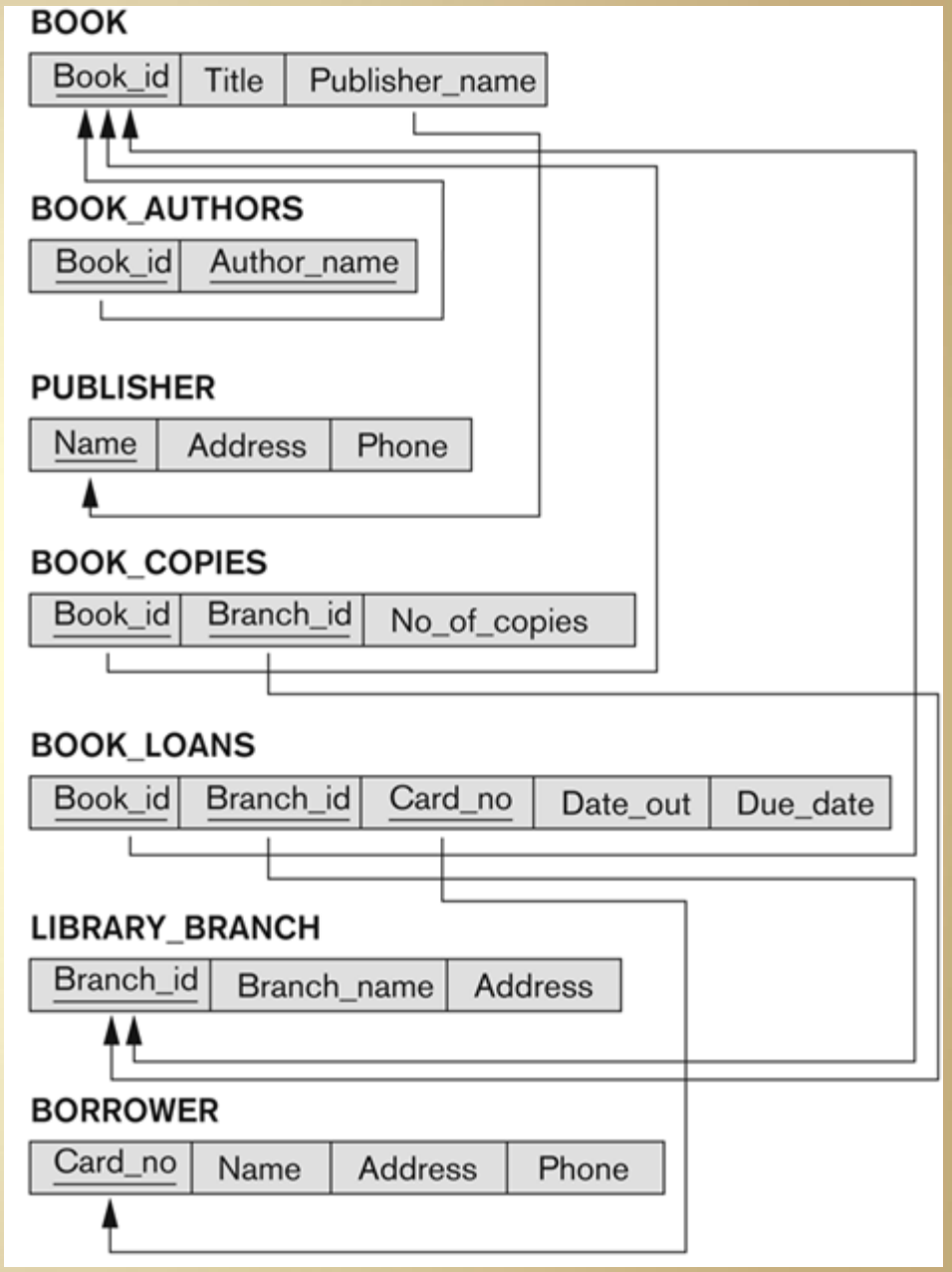
ENROLL(Student, Course, Semester, Grade)

BOOK_ADOPTION(Course, Semester, ISBN)

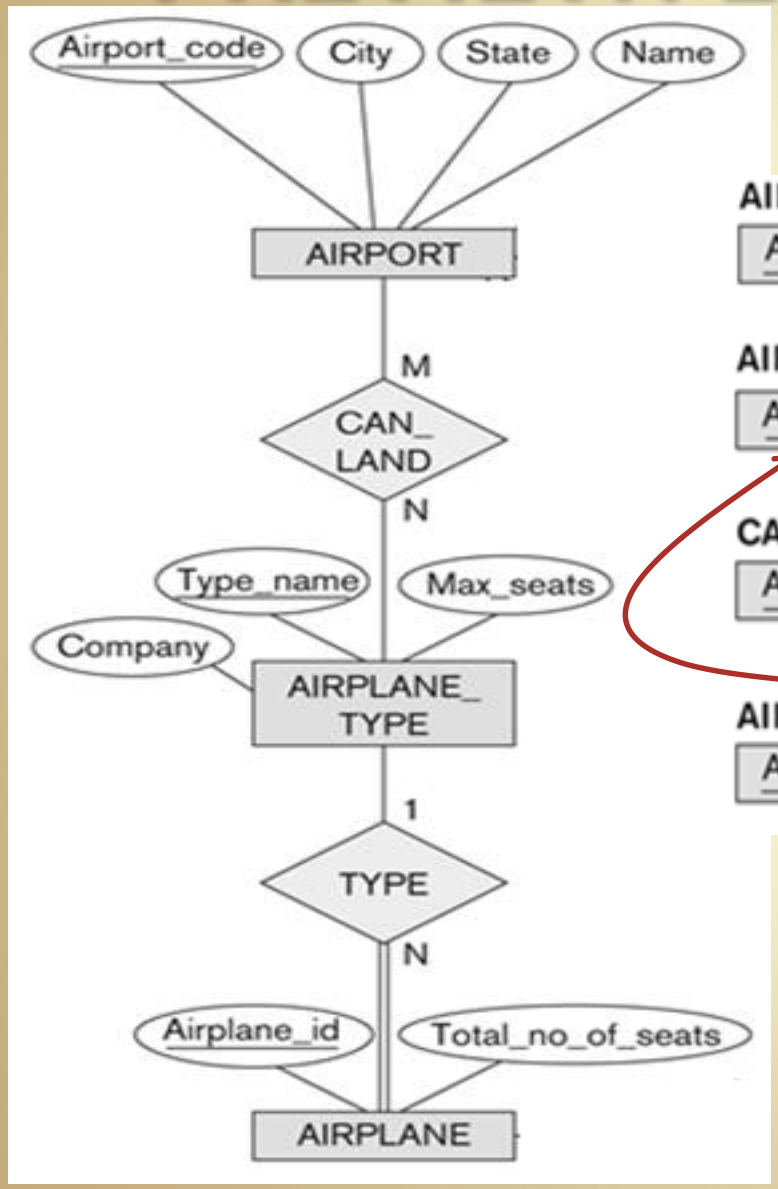
TEXTBOOK(ISBN, Title, Publisher, Author)

REVIEW

Reverse engineer this relational schema to find an equivalent ER schema.



PREVIEW: ER to Relational



AIRPORT

Airport_code	Name	City	State

AIRPLANE_TYPE

Airplane_type_name	Max_seats	Company

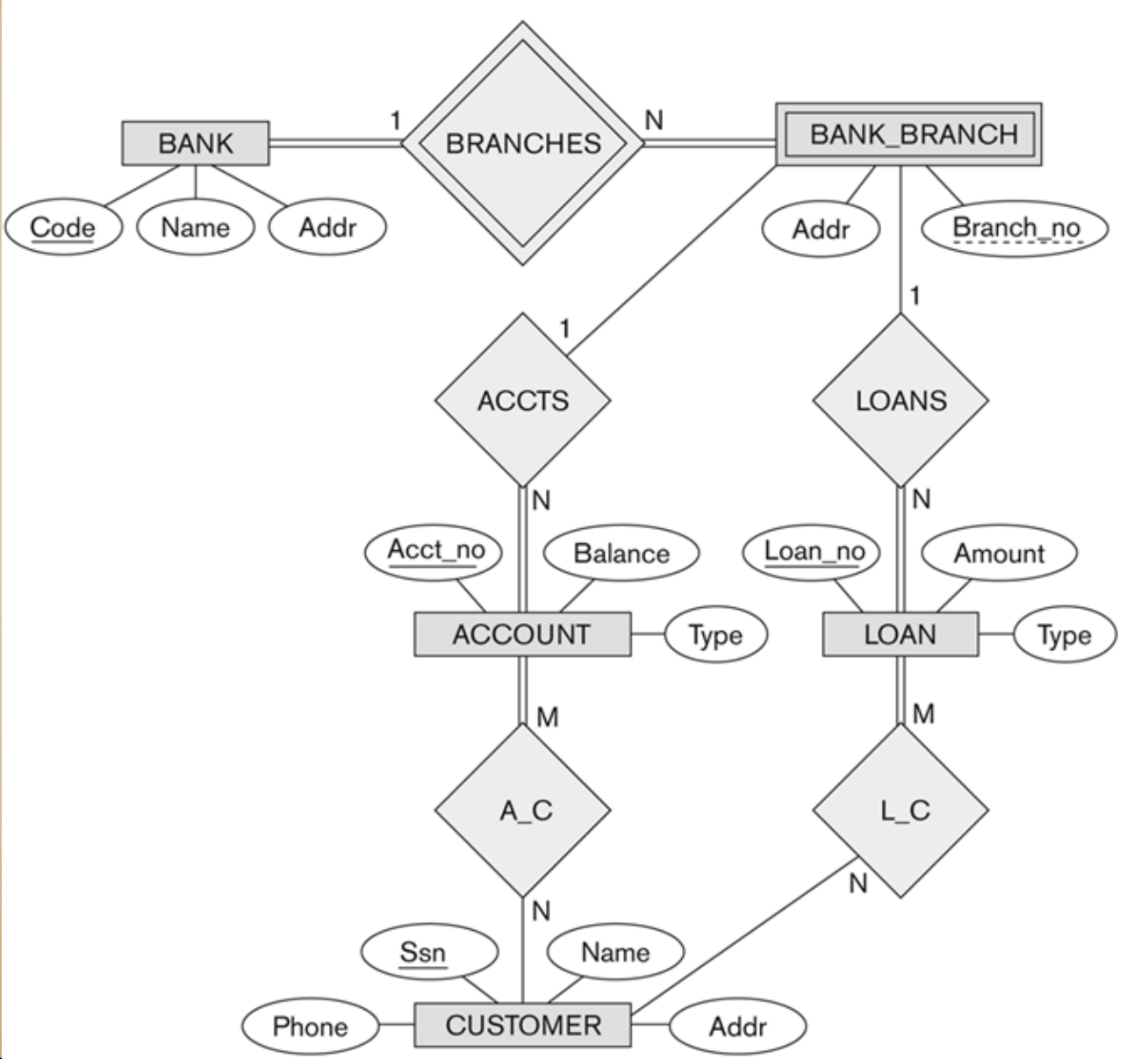
CAN_LAND

Airplane_type_name	Airport_code

AIRPLANE

Airplane_id	Total_number_of_seats	Airplane_type

EER Bank Schema

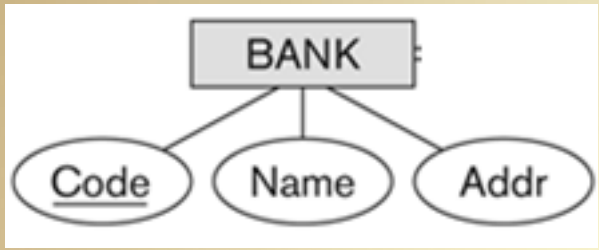


Step 1: Regular Entities

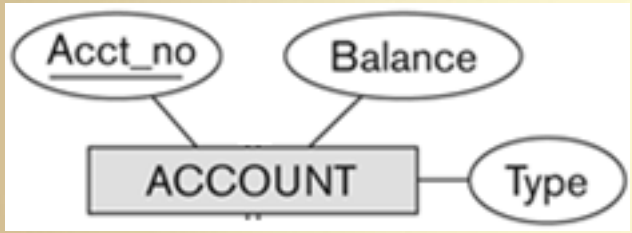
- Regular entity types become relations
  - include all simple attributes
  - include only components of compound attributes
  - keys become primary keys
  - if multiple keys (candidates) select a primary key

CUSTOMER(Ssn, Name, Addr, Phone)

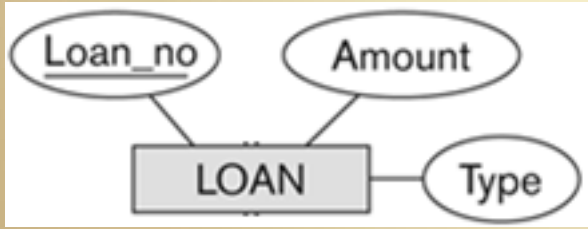
Step 1: Regular Entities



BANK(Code, Name, Addr)



ACCOUNT(Acct_no, Type, Balance)



LOAN(Loan_no, Type, Amount)

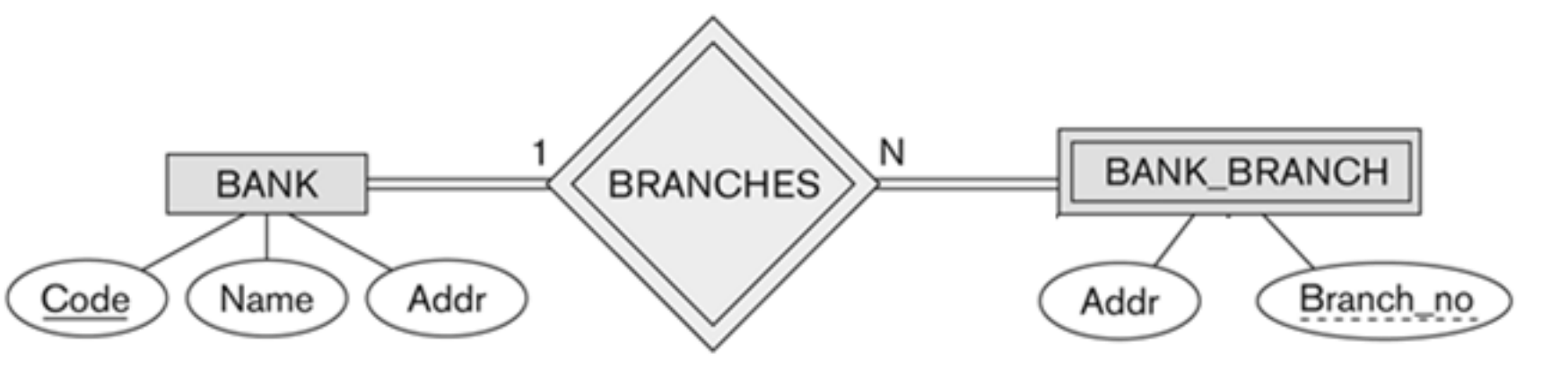
Step 2: Weak Entities

- Weak entity types become relations
  - include all simple attributes
  - include only components of compound attributes
  - create a primary key from partial key
    and key of owning entity type (through identifying relationship)
  - attributes acquired through identifying relationship
    become a foreign key*

* typically, deletions and insertions will be propagated through this foreign key

Step 2: Weak Entities

- Weak entity types become relations



BANK_BRANCH(Bank code, Branch No, Addr)

FK

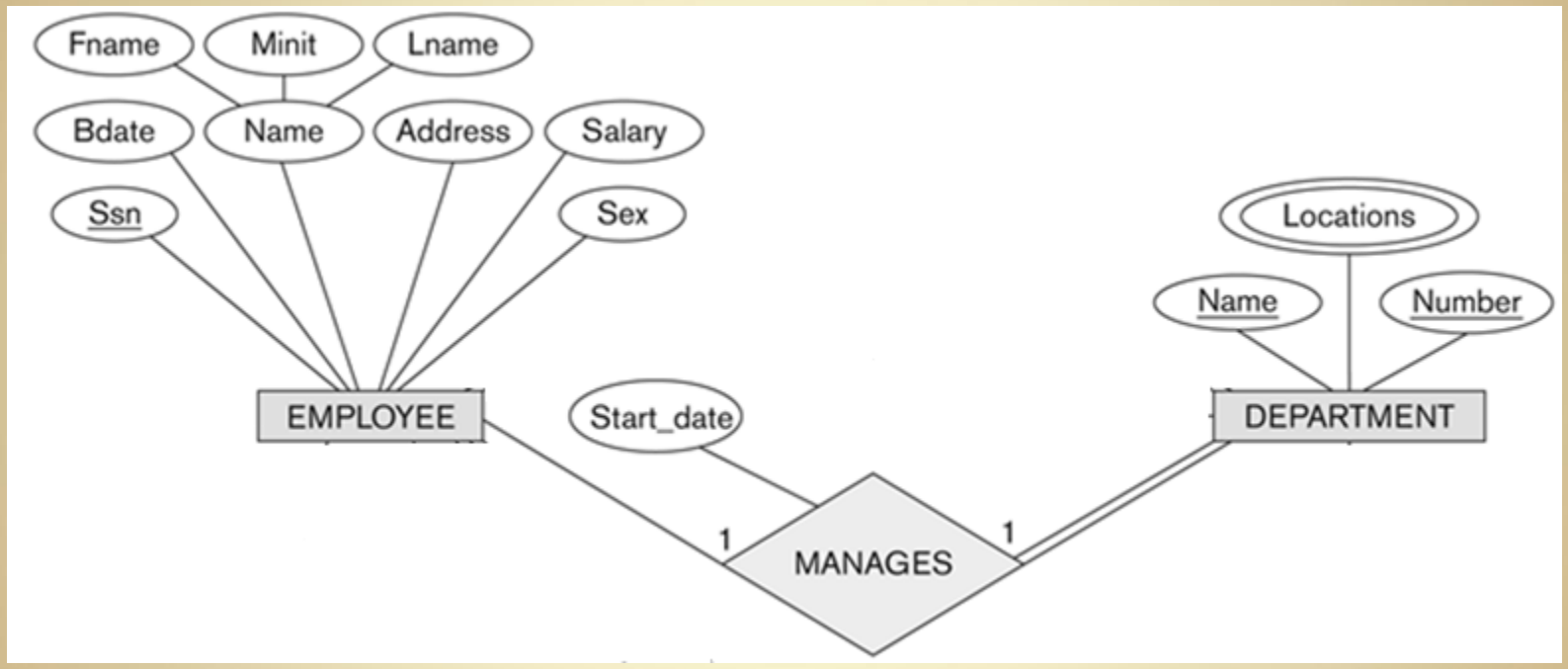
BANK(Code, Name, Addr)

Step 3: Binary 1:1 Relationships

- Approach 1: Foreign Key
  - Chose one of the related entity types to hold the relationship (chose one with total participation, if possible)
  - add FK to other relation
  - move all relationship attributes to this relation
  - *this approach is preferable, except as noted below*
- Approach 2: Merged Relation
  - combine the relations for the related entities into a single relation
  - *use only when both participations are total*
- Approach 3: Separate Relation
  - same as binary M:N relationship (see step 5)
  - *not generally a good option*

Step 3: Binary 1:1 Relationships

- Approach 1: Foreign Key



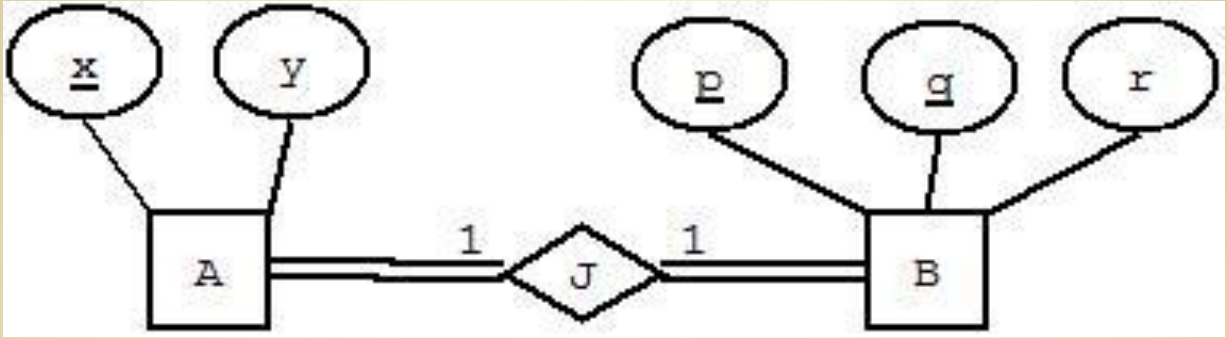
EMPLOYEE(Ssn, Name, …)

FK

DEPARTMENT(Name, Number, Mgr, Mgr_start_date)

Step 3: Binary 1:1 Relationships

- Approach 2: Merged Relation



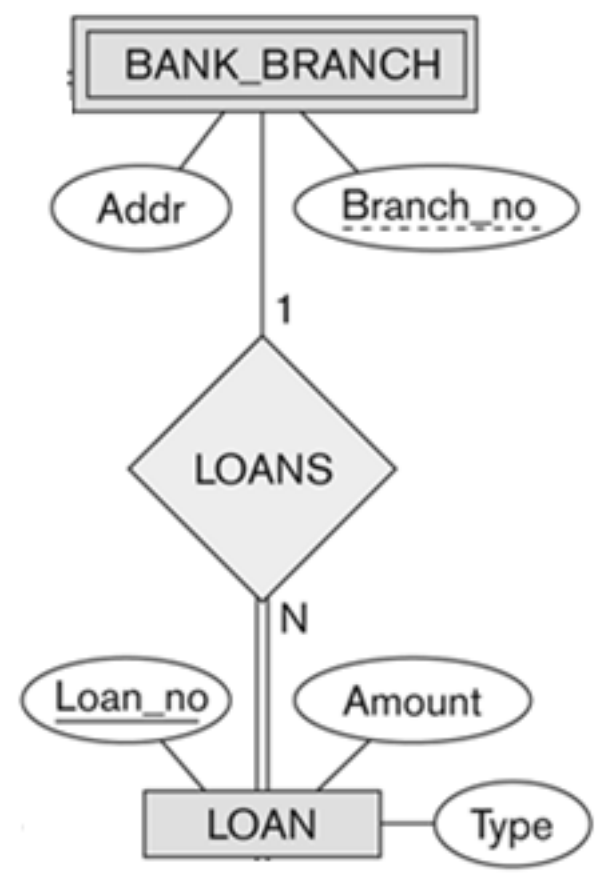
AJB(x, y, p, q, r)

or

AJB(x, y, p, q, r)

Step 4: Binary 1:N Relationships

- 1:N Relationships become foreign key at N side
  - any relationship attributes also go to N side

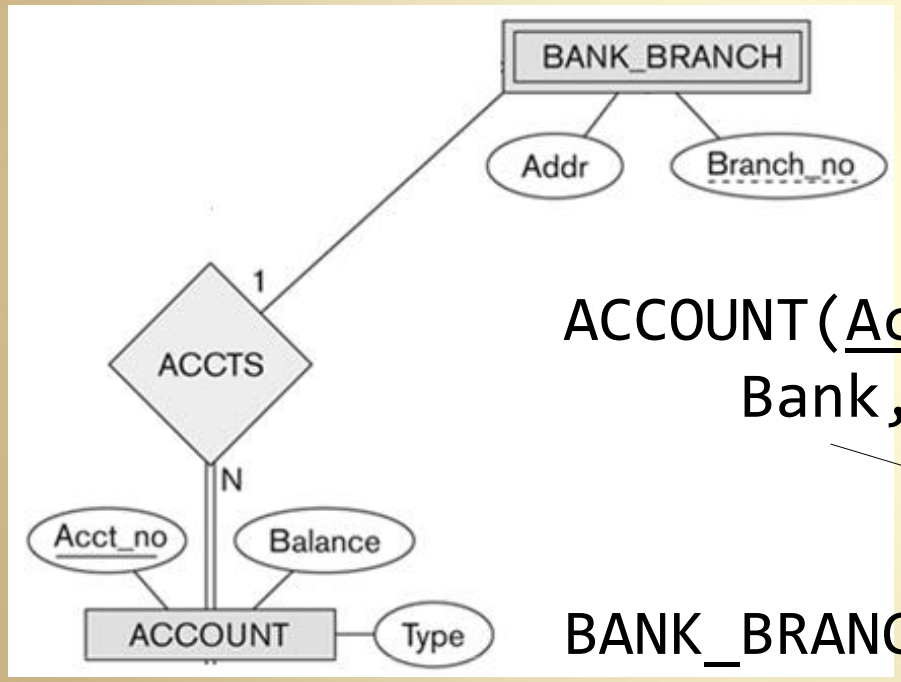


LOAN(Loan_no, Type, Amount, Bank, Branch)

BANK_BRANCH(Bank_code, Branch_No, Addr)

Step 4: Binary 1:N Relationships

- 1:N Relationships become foreign key at N side
  - any relationship attributes also go to N side

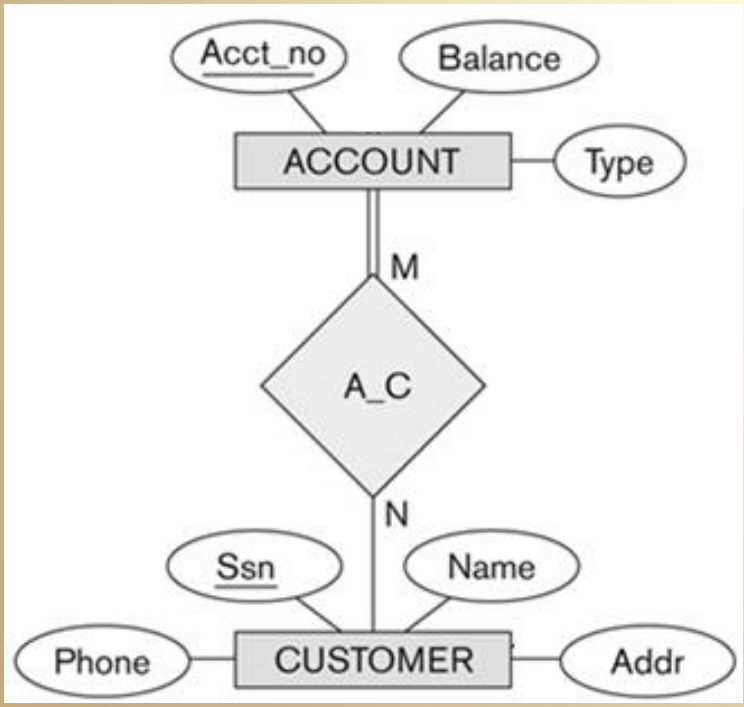


ACCOUNT(Acct_no, Type, Balance, Bank, Branch)

BANK_BRANCH(Bank_code, Branch_No, Addr)

Step 5: Binary M:N Relationships

- M:N Relationships must become a new relation
  - contains FKs to both related entities
  - combined FKs become PK for new relations
  - relationship attributes go in new relation



CUSTOMER(Ssn, Name, Addr, Phone)

A_C(Acct, Cust)

ACCOUNT(Acct_no, Type, Balance, Bank, Branch)

Step 6: Multivalued Attributes

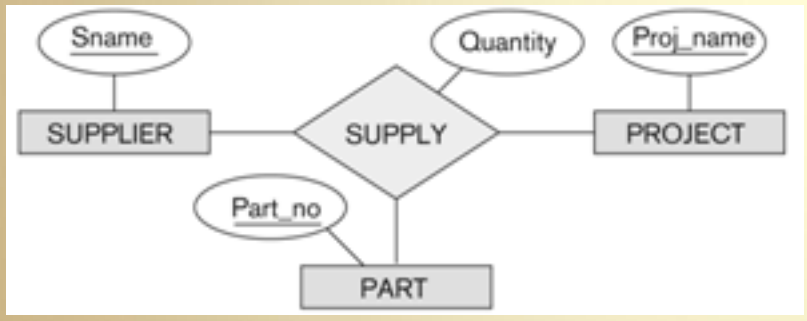
- Multivalued attributes must become new relations
  - FK to associated entity type
  - PK is whole relation

DEPARTMENT(Name, Number, Mgr, Mgr_start_date)

DEPT_LOCATIONS(DName, Dno, Location)

Step 7: N-ary Relationships

- Non-Binary Relationships become new relations
  - FKs to all participating entity types
  - Combine FKs to make a PK
    (exclude entities with max participation of 1)
  - Include any relationship attributes



SUPPLIER(SName)

PROJECT(Proj_name)

PART(Part_no)

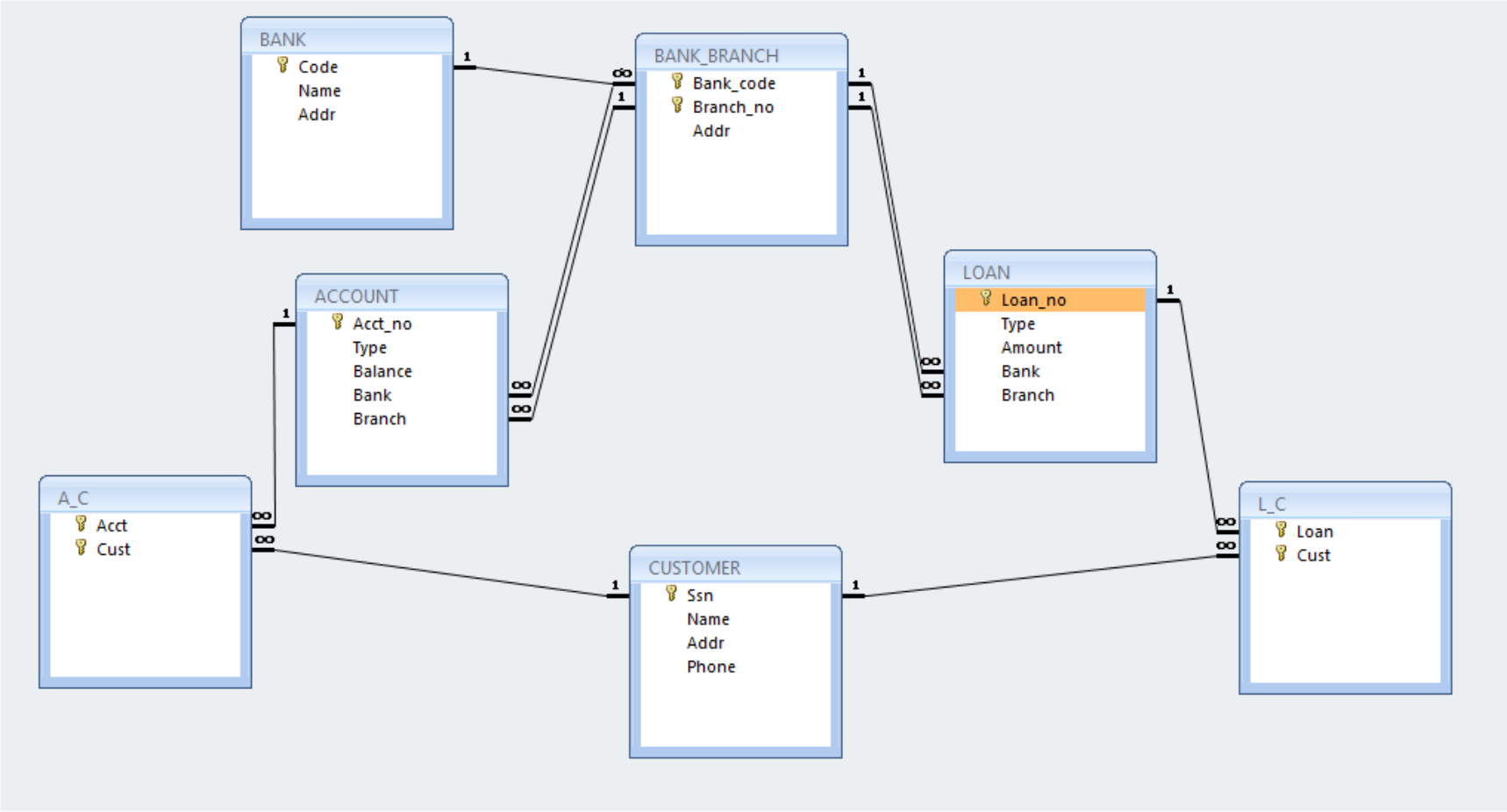
SUPPLY(SName, PName, Part, Quantity)

Completed Bank Schema

CUSTOMER(Ssn, Name, Addr, Phone)
BANK(Code, Name, Addr)
ACCOUNT(Acct no, Type, Balance, Bank, Branch)
LOAN(Loan no, Type, Amount, Bank, Branch)
BANK_BRANCH(Bank code, Branch No, Addr)
A_C(Acct, Cust)
L_C(Loan, Cust)

BANK_BRANCH(Bank_code) refers to BANK
LOAN(Bank,Branch) refers to BANK_BRANCH
ACCOUNT(Bank,Branch) refers to BANK_BRANCH
A_C(Acct) refers to ACCOUNT
A_C(Cust) refers to CUSTOMER
L_C(Loan) refers to LOAN
L_C(Cust) refers to CUSTOMER

Bank Schema: MS Access

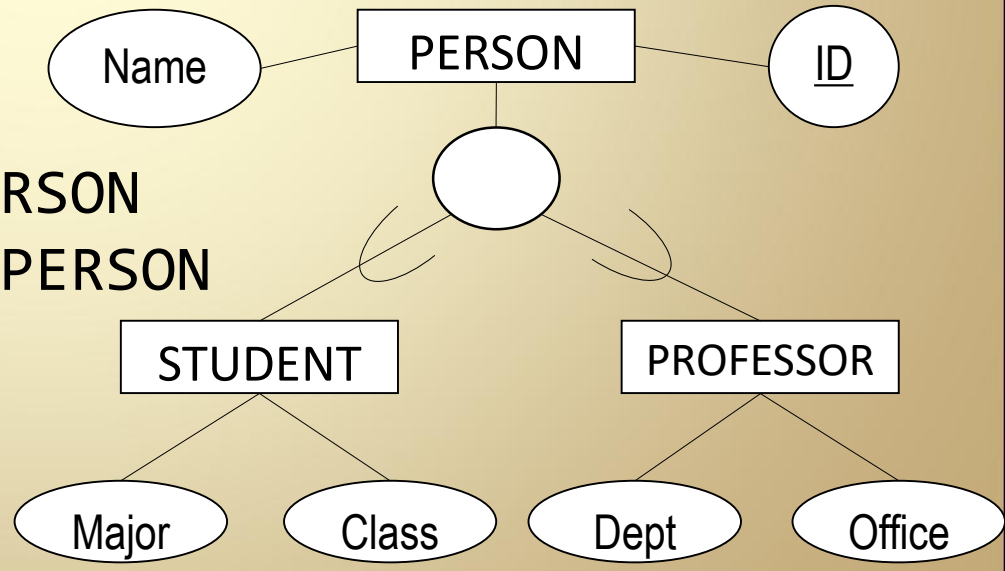


Step 8: Inheritance

- Option a: Each entity type becomes a relation
  - all have same PK (from superclass)
  - PKs in subclasses are FKs to superclass
  - *most general solution*

```
PERSON(ID, Name)
STUDENT(ID, Major, Class)
PROFESSOR(ID, Dept, Office)
```

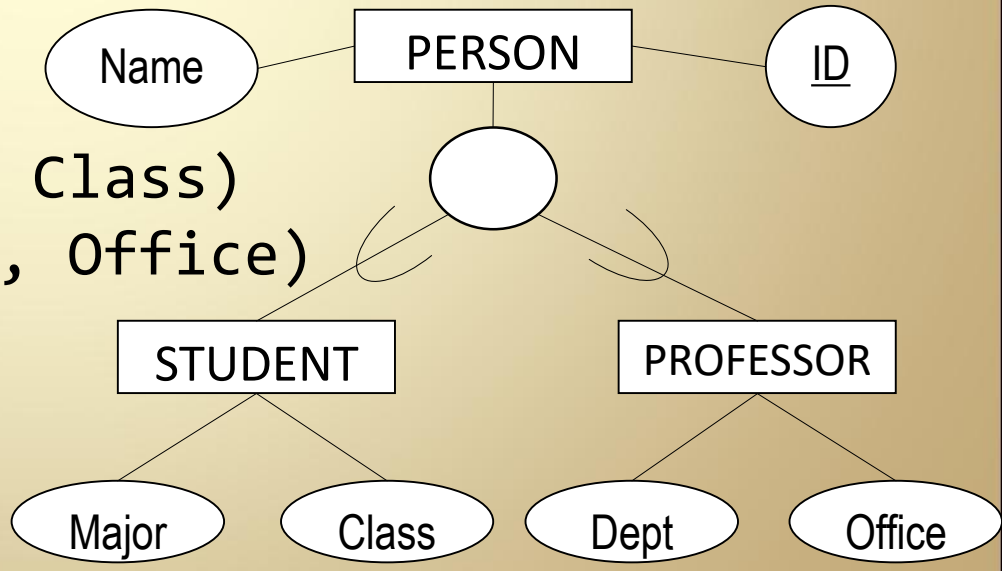
```
STUDENT(ID) refers to PERSON
PROFESSOR(ID) refers to PERSON
```



Step 8: Inheritance

- Option b: Each subclass becomes a relation
  - all have same PK (from superclass)
  - each relation gets all superclass attributes
  - *restriction: only works for covering inheritance*
  - *problem: need to join tables to find all PERSONs*

```
STUDENT(ID, Name, Major, Class)
PROFESSOR(ID, Name, Dept, Office)
```

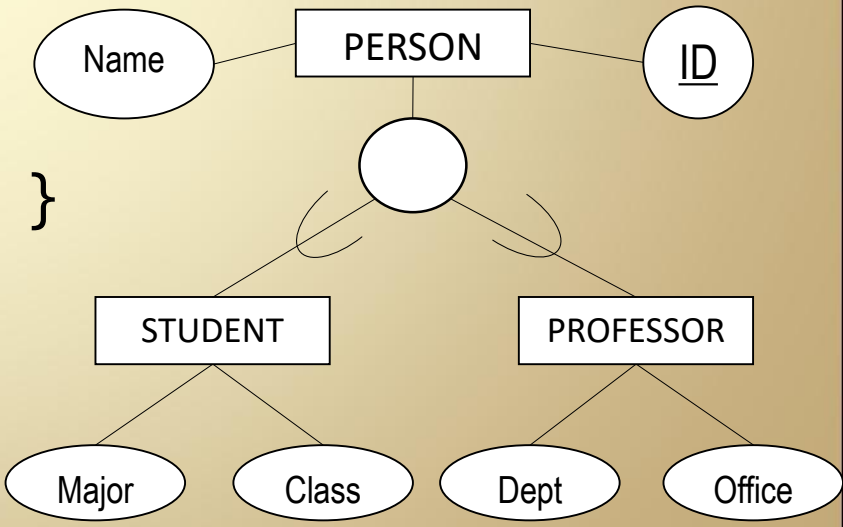


Step 8: Inheritance

- Option c: Single relation with a type discriminator
  - PK from superclass
  - all attributes from all classes
  - *restriction: only works for disjoint inheritance*
  - *problem: lots of NULL values*

```
PERSON(ID, Classifier, Name,
     Major, Class, Dept, Office)
```

```
Classifier ϵ { ‘S’, ‘P’, ‘N’ }
```

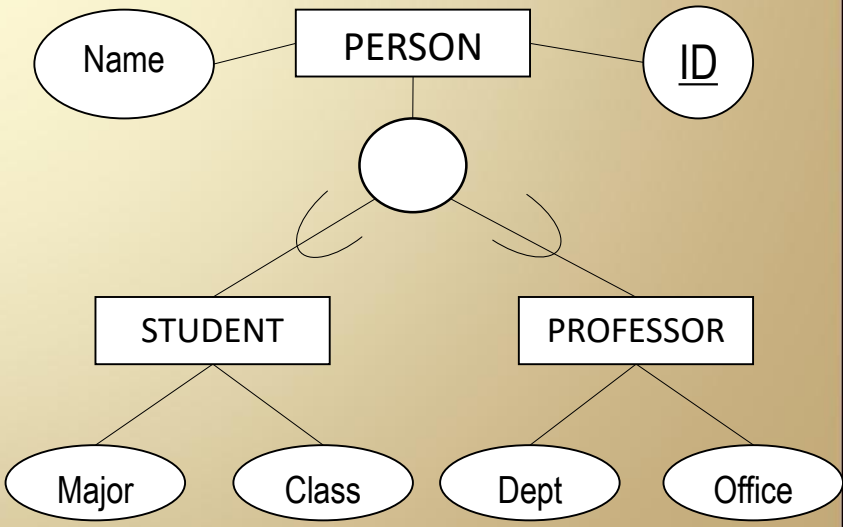


Step 8: Inheritance

- Option d: Single relation with multiple discriminators
  - PK from superclass
  - all attributes from all classes
  - *works for overlapping inheritance*
  - *problem: lots of NULL values*

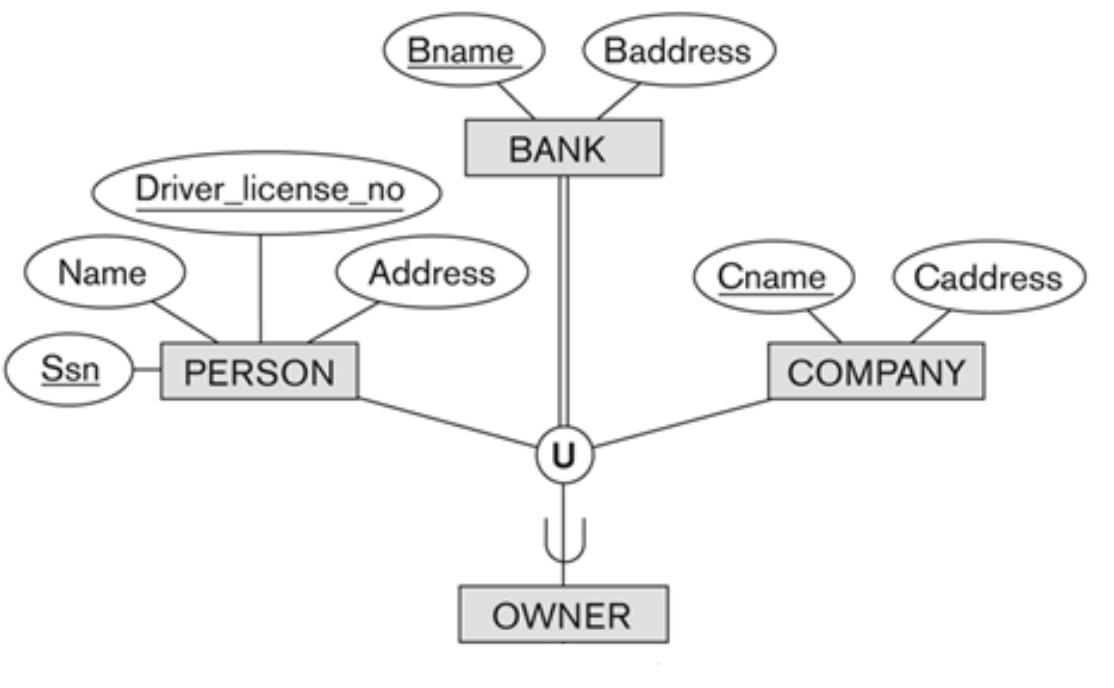
```
PERSON(ID, isStudent, isProfessor,
Name, Major, Class, Dept, Office)
```

```
dom(isStudent) = Boolean
dom(isProfessor) = Boolean
```



Step 9: Unions

- Union types become a new relation of surrogate keys
  - surrogate keys are added to all defining classes
  - attributes of the union type go in the new relation



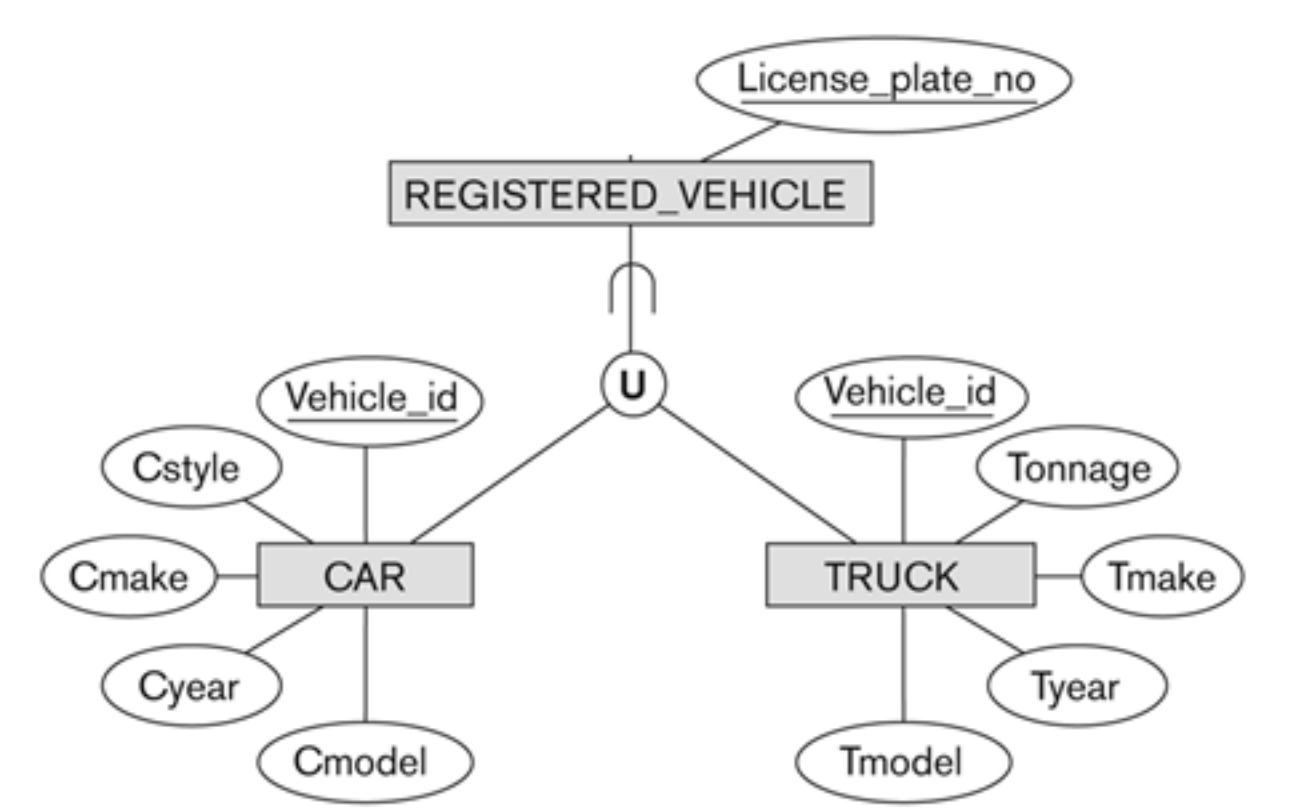
add surrogate key to OWNER

Step 9: Unions

PERSON(Driver_license_no, Ssn, Name,
Address, Owner_id)
BANK(Bname, Baddress , Owner_id)
COMPANY(Cname, Caddress , Owner_id)
OWNER(ID)

PERSON(Owner_id) refers to OWNER
BANK(Owner_id) refers to OWNER
COMPANY(Owner_id) refers to OWNER

Step 9: Unions



add surrogate key to REGISTERED_VEHICLE

Step 9: Unions

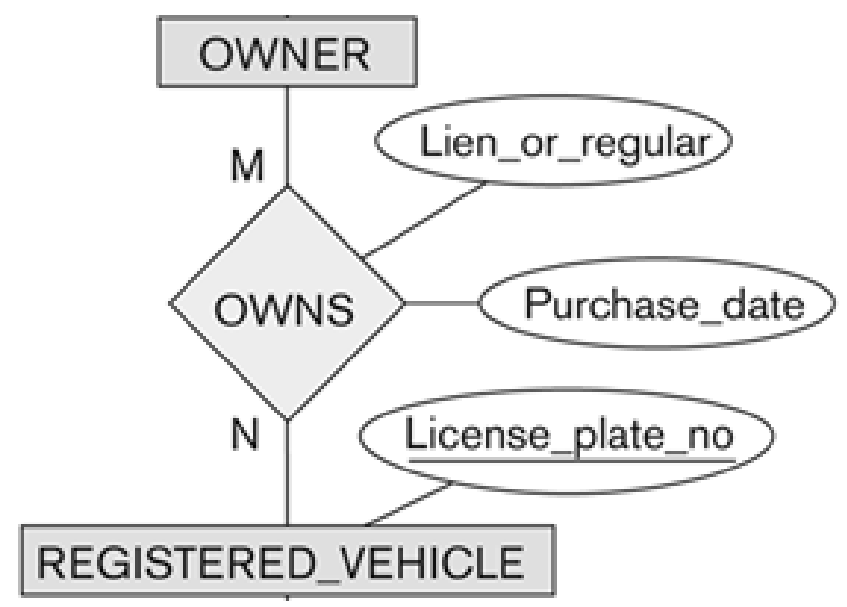
CAR(Vehicle_id, Cstyle, Cmake,
Cyear, Cmodel)
TRUCK(Vehicle_id, Tonnage, Tmake,
Tyear, Tmodel)
REGISTERED_VEHICLE(Vehicle_id, License_plate_no)

CAR(Vehicle_id) refers to REGISTERED_VEHICLE
TRUCK(Vehicle_id) refers to REGISTERED_VEHICLE

in this case, we don't need to invent a surrogate key, since the domains of CAR keys and TRUCK keys are the same (and non-overlapping)

Step 9: Unions

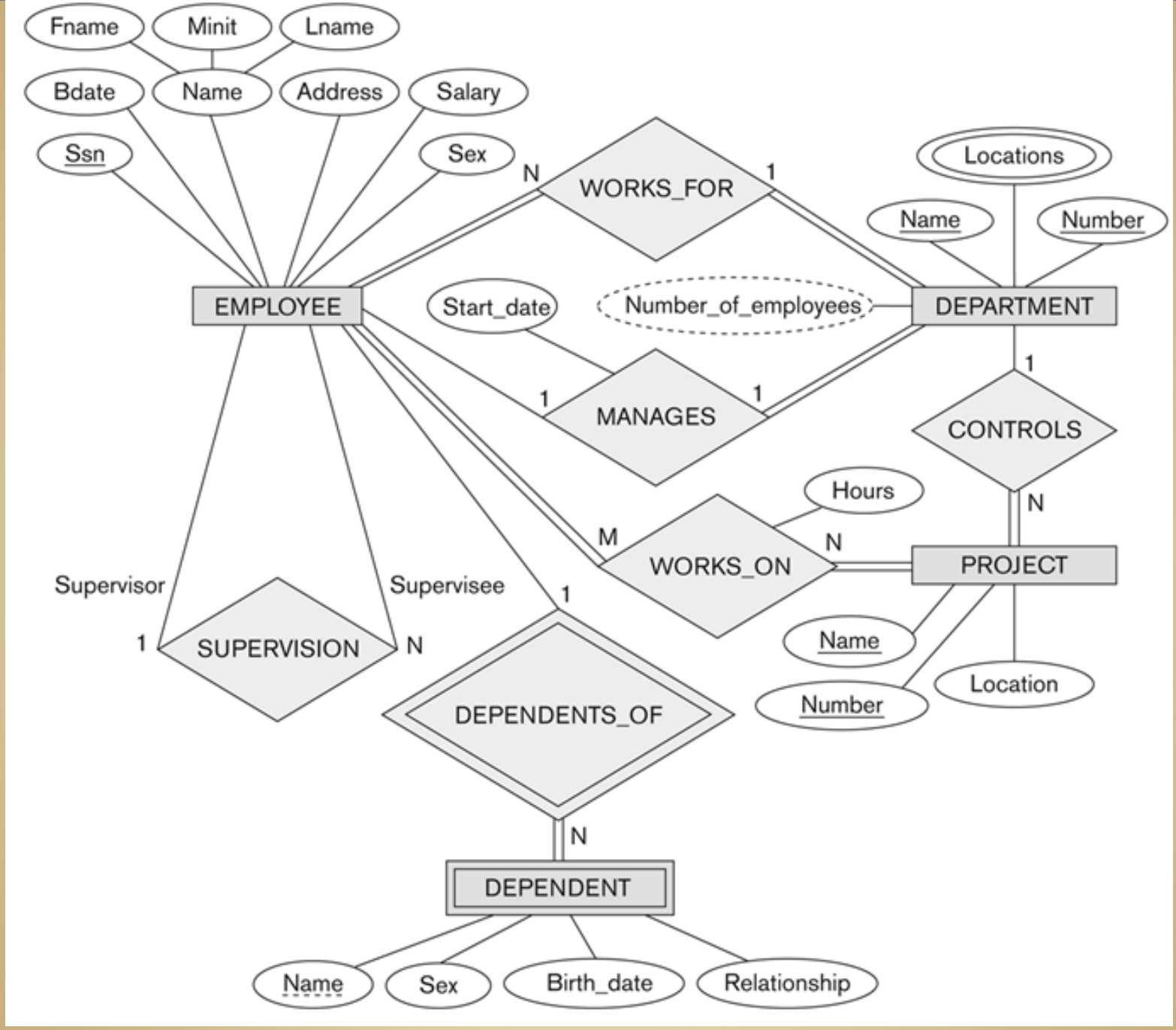
OWNS relation uses the surrogate keys

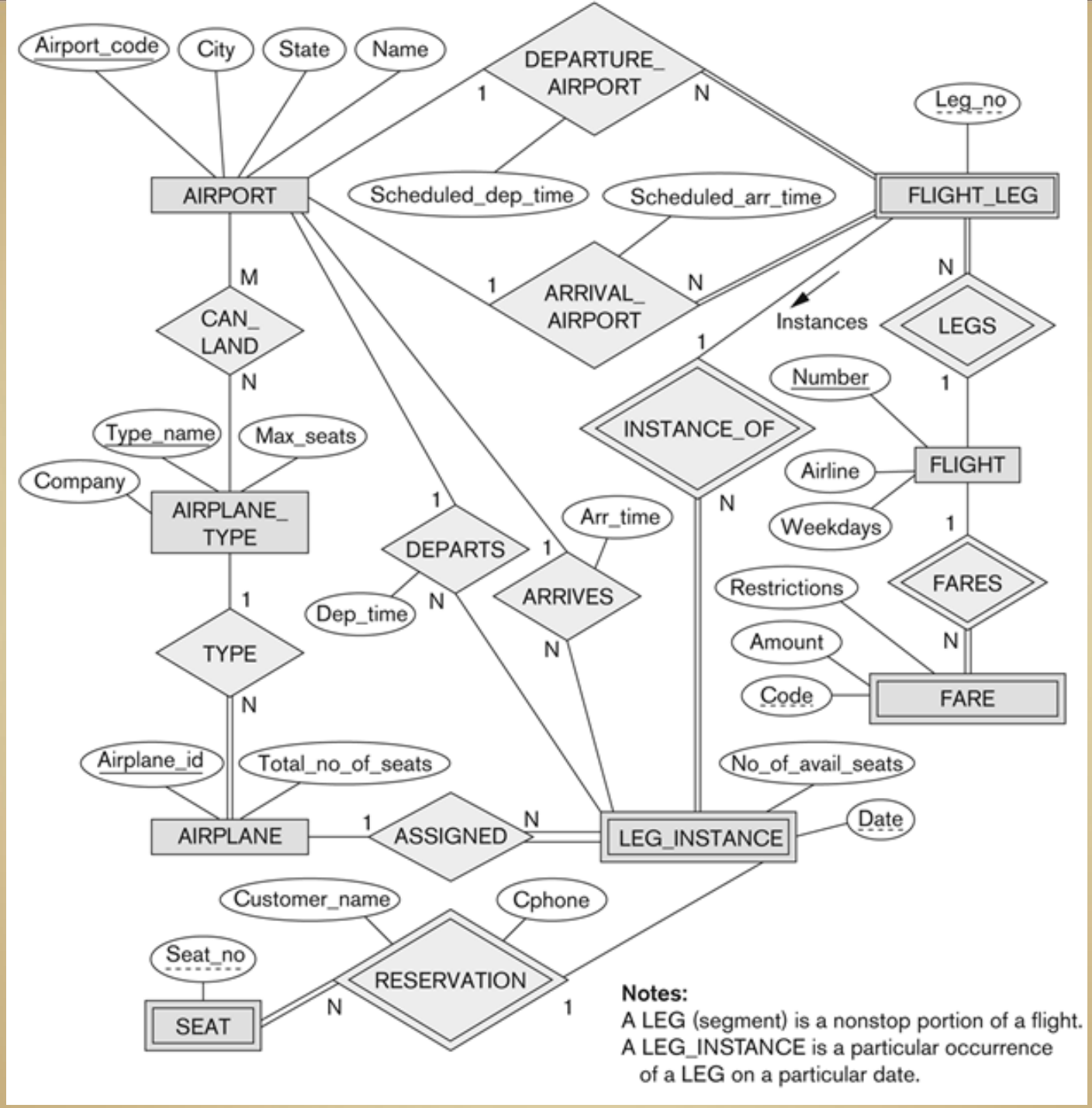


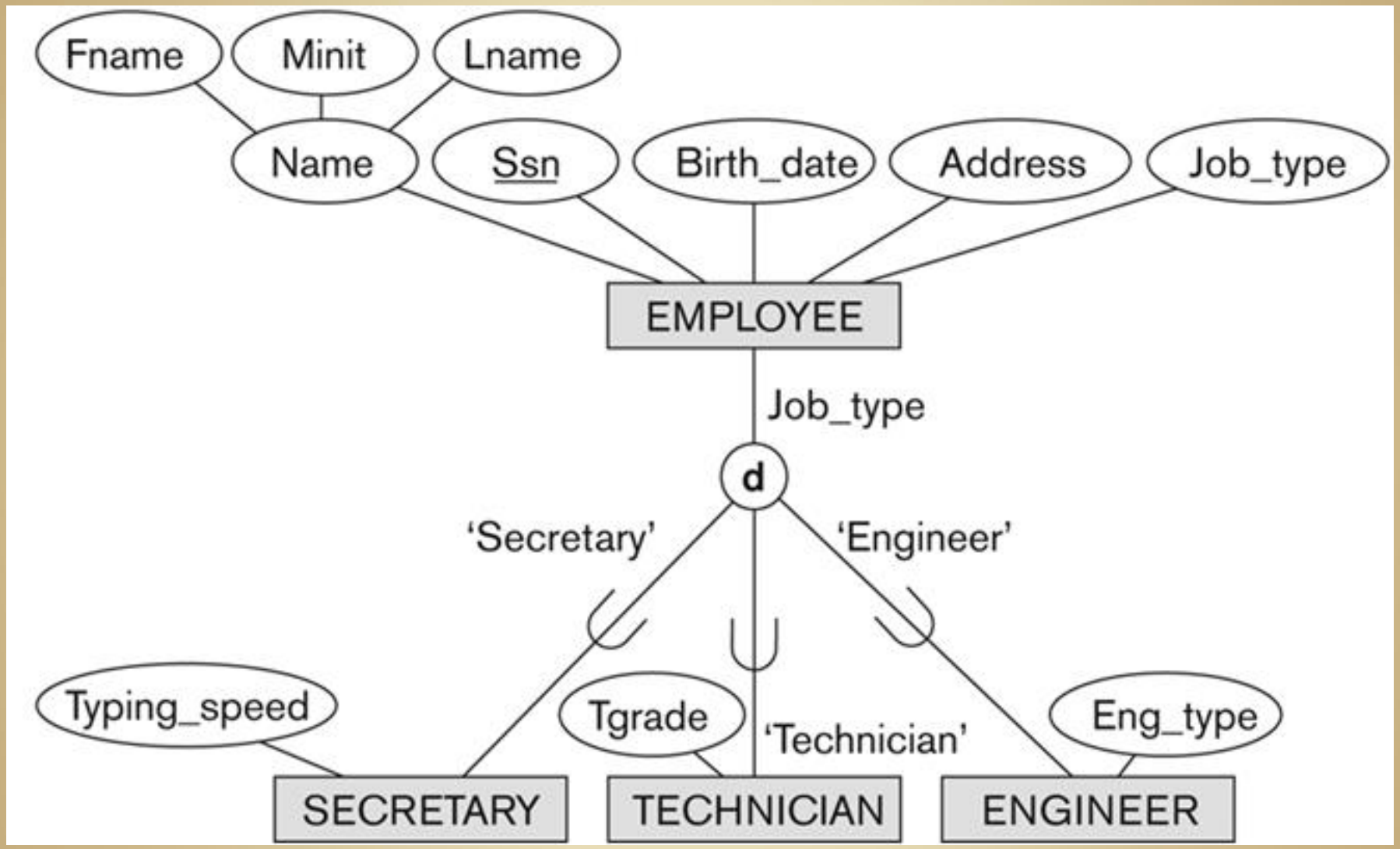

```
OWNS(Owner_id, Vehicle_id,
     Purchase_date, Lien_or_regular)

OWNS(Owner_id) refers to OWNER
OWNS(Vehicle_id) refers to REGISTERED_VEHICLE
```

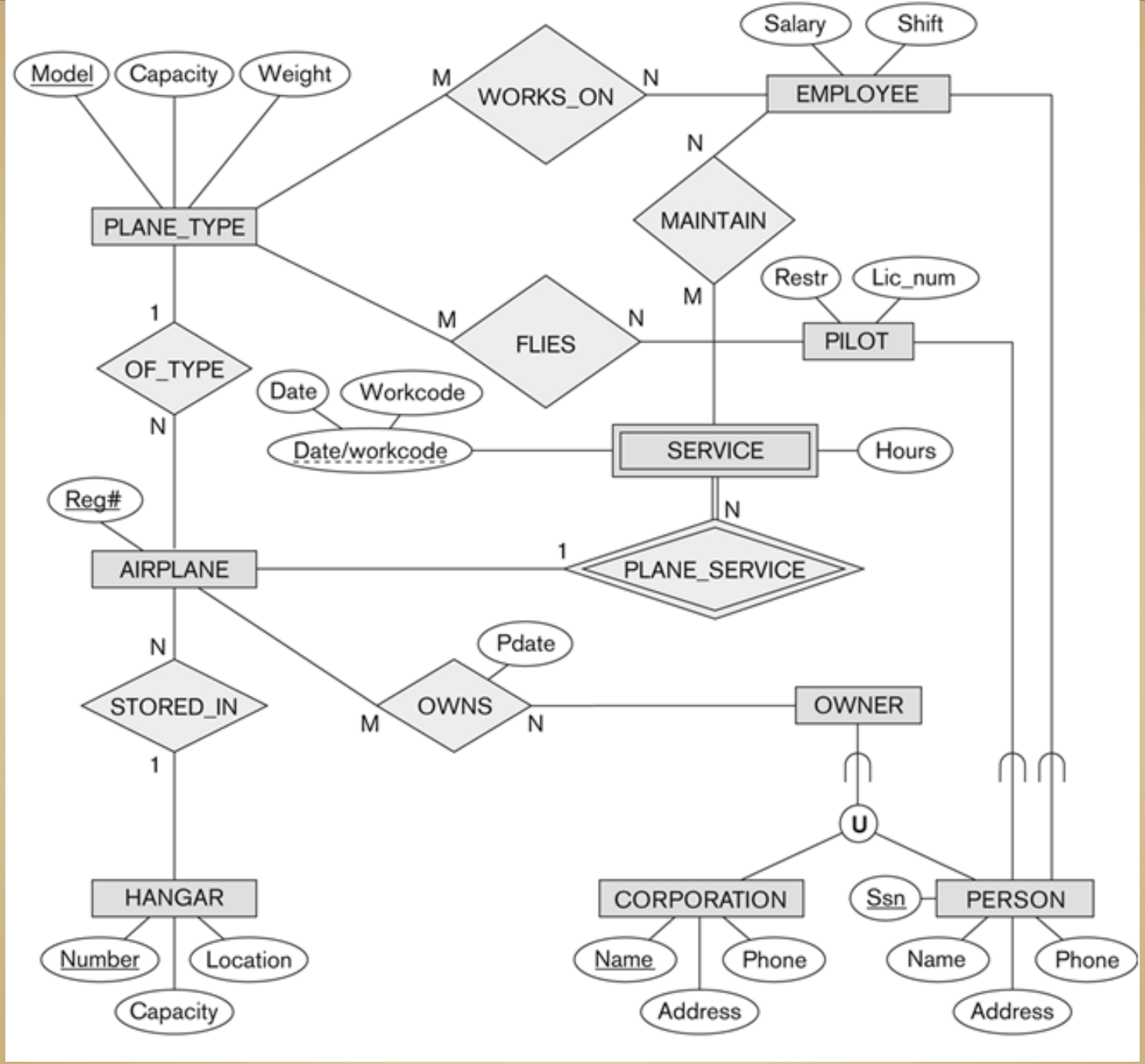
EXERCISES

- Create relational schema from the following:


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
MANAGES
1
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship


Airport_code
City
State
Name
DEPARTURE_ AIRPORT
1
N
Leg_no
AIRPORT
Scheduled_dep_time
Scheduled_arr_time
FLIGHT_LEG
1
ARRIVAL_ AIRPORT
N
Instances
N
M
CAN_ LAND
LEGS
1
N
1
Number
INSTANCE_OF
Type_name
Max_seats
Airline
FLIGHT
Company
AIRPLANE_ TYPE
1
N
Arr_time
Weekdays
1
1
DEPARTS
ARRIVES
FARES
Restrictions
Dep_time
N
1
N
Amount
N
TYPE
Code
FARE
N
Airplane_id
Total_no_of_seats
No_of_avail_seats
AIRPLANE
1
ASSIGNED
N
LEG_INSTANCE
Date
Customer_name
Cphone
Seat_no
RESERVATION
SEAT
N
1
Notes:
A LEG (segment) is a nonstop portion of a flight.
A LEG_INSTANCE is a particular occurrence of a LEG on a particular date.


Fname
Minit
Lname
Name
Ssn
Birth_date
Address
Job_type
EMPLOYEE
Job_type
d
'Secretary'
'Engineer'
'Technician'
Typing_speed
Tgrade
Eng_type
SECRETARY
TECHNICIAN
ENGINEER


Salary
Shift
Model
Capacity
Weight
M
WORKS_ON
N
EMPLOYEE
N
MAINTAIN
PLANE_TYPE
M
Restr
Lic_num
1
M
FLIES
N
PILOT
OF_TYPE
Date
Workcode
N
Date/workcode
SERVICE
Hours
Reg#
N
1
AIRPLANE
PLANE_SERVICE
N
Pdate
STORED_IN
M
OWNS
N
OWNER
1
U
HANGAR
CORPORATION
Ssn
PERSON
Number
Location
Name
Phone
Name
Phone
Capacity
Address
Address

Preview: Queries

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Preview: Queries

- Which employee has no supervisor?
- Which employees are supervised by Franklin Wong?
- Which employees have dependents?
- Which employees have daughters?
- Which employees in department 5 work more than 10 hours/week on ProductX?
- Which department has the highest paid manager?
- What is the average salary of all department managers?
- Which employees don't have daughters?