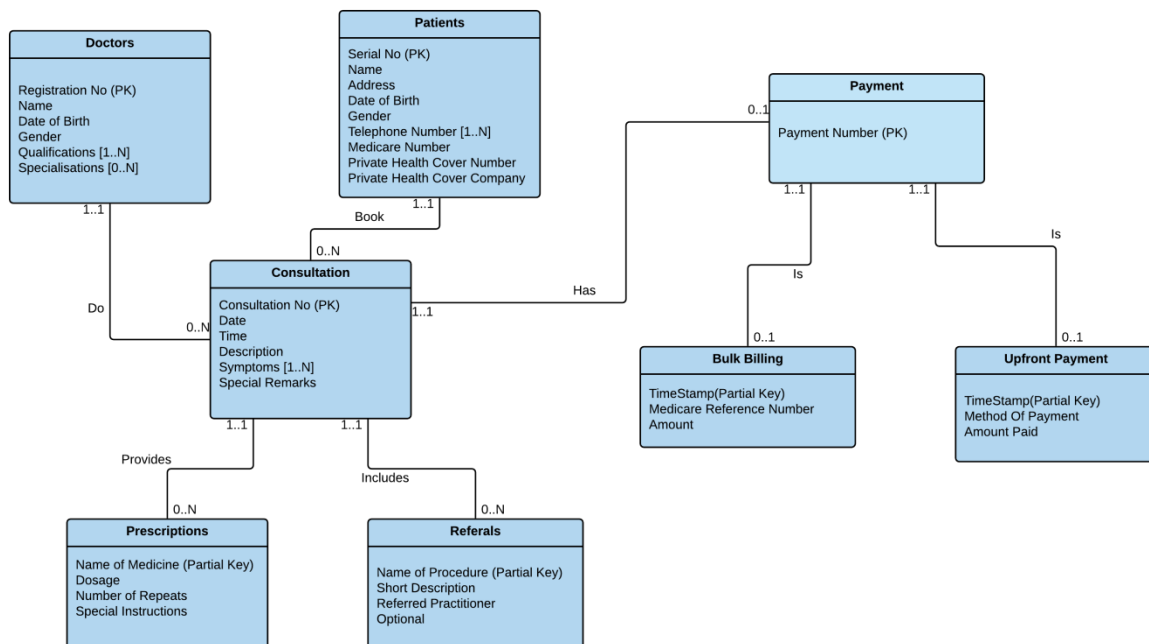


Database Concepts : Assignment – 1

2 Part A: Entity-Relationship Modelling

2.1 : Designing an Entity-Relationship Model :-

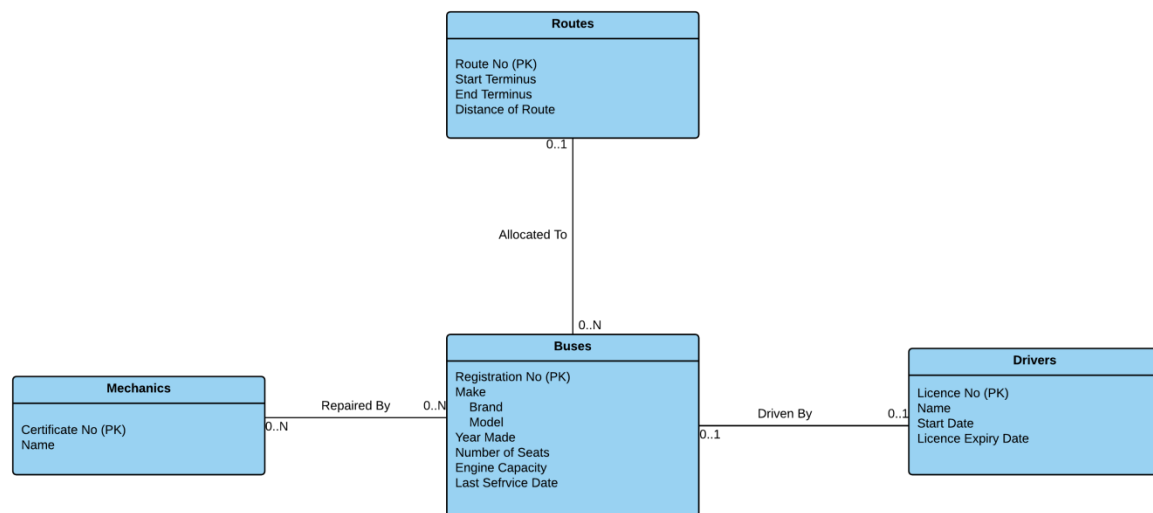


Assumptions :-

- 1) It may be possible that Patient have not been visited any doctor or have not booked any Consultation.
- 2) It may be possible that there are no Symptoms and No Prescriptions or Referrals are included in Consultation.
- 3) It is assumed that only one payment number is generated in case of Upfront & Bulk billing combined. And then two entries are made, one in each entity - Upfront Payment and Bulk billing with same Payment Number.
- 4) Assuming each doctor has at least one qualification but not all doctors have at least one specialization.

2.2 : Designing and refining an Entity-Relationship Model :-

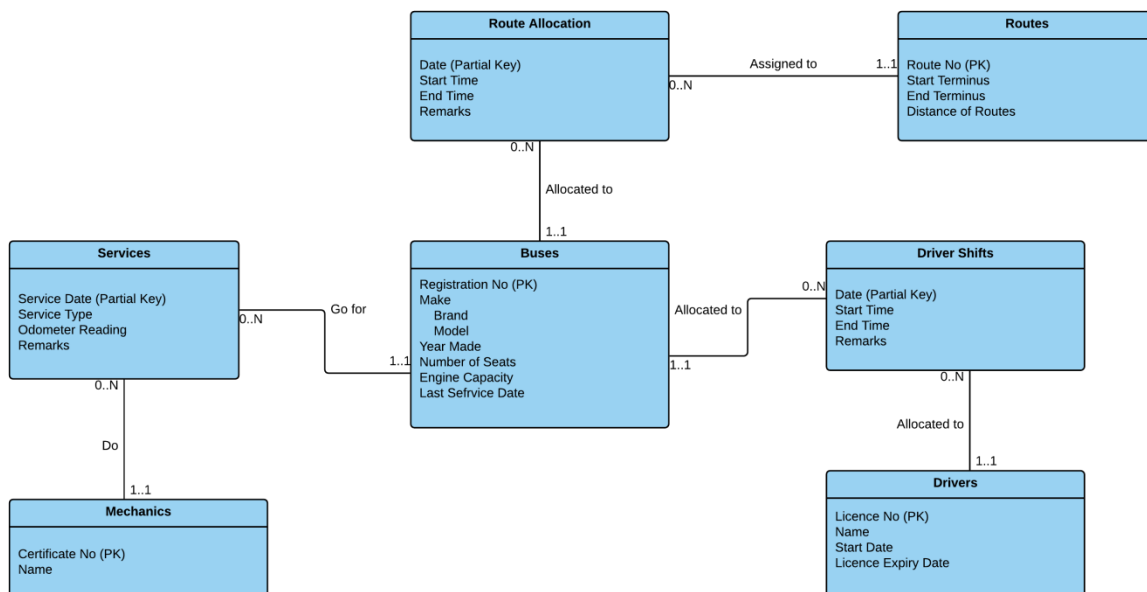
A)



Assumptions :-

- 1) Bus Can be allocated to zero (If in service) or one Route and Route can be allocated to zero(If no bus assigned to route) or more buses.
- 2) Bus Can be allocated to zero (If in service) or one Drivers and Driver can be allocated to zero(If on Leave) or one buses.
- 3) Bus Can be repaired by zero (If bus is not repaired even a once) or more Mechanics and Mechanic can repair zero (If Mechanic has not repaired any bus yet) or more buses.

B)



Assumptions :-

- 4) Bus Can be allocated to zero (If in service) or more Routes (Same bus can have different routes during different time) and Route can be allocated to zero (If no bus assigned to route) or more buses.
- 5) Bus Can be allocated to zero (If in service) or more Drivers and Driver can be allocated to zero (If on Leave) or more buses.
- 6) Assuming driver only work one shift per day.

2.3 : Mapping an ER Model to Relational Database Schema

Step – 1: Map Strong Entities :- Create a relation for each entity type with their primary key as primary key of new relation. Separate the components of composite attributes as individual attributes, if any.

Author (Email, Name, Address, Telephone)

Publisher (Name, Address, URL, ABN)

Book (ISBN, Title, Edition, ListPrice)

Warehouse(Code, Address)

ShoppingCart (CartID, TimeStamp)

Customer (Email, Name, Address)

Step – 2: Map Weak Entities :- Create a new relation with all attributes of weak entity and add primary key of owner entity in new relation as a foreign key. This primary key and partial key of weak entity will together form primary key of new relation.

No Weak Entities.

Step – 3: Map 1:1 Relationships :- Copy the primary key of opposite entity into relation which has mandatory-participating entity. Add attributes of relation as a simple attribute in new relation, if any.

No 1:1 Relationship.

Step – 4: Map 1:N Relationships :- Copy the primary key of opposite entity into relation with N-side entity. Add attributes of relation as a simple attribute in new relation, if any.

ShoppingCart (CartID, TimeStamp, Email*)

Step – 5: Map M:N Relationships :- Create New Relation including Primary keys of each participating Entities. Both these together form a primary key of new relation. Add attributes of relation as a simple attribute in new relation, if any.

Author-Book (Email*, ISBN*)

Book-Publisher (ISBN*, Name*)

Book-Warehouse (ISBN*, Code*, StockQty)

Book-ShoppingCart (ISBN*, CardID*, BuyPrice, Qty)

Step – 6: Multi-Valued Attributes :- Create a new relation for each multi-valued attribute including the corresponding attribute along with the primary key of owner entity. Both these together form a primary key of new relation.

Author-Telephone(Email*, Telephone)

Step – 7: Map higher-degree Relationships :- Create a new relation with primary keys from all participating entities. Together, They form the primary key of the new relation.

No higher-degree Relationships.

Complete Relational Model is as follows :-

Author (Email, Name, Address)

Publisher (Name, Address, URL, ABN)

Book (ISBN, Title, Edition, ListPrice)

Warehouse (Code, Address)

ShoppingCart (CartID, TimeStamp, Email*)

Customer (Email, Name, Address)

Author-Book (Email*, ISBN*)

Book-Publisher (ISBN*, Name*)

Book-Warehouse (ISBN*, Code*, StockQty)

Book-ShoppingCart (ISBN*, CardID*, BuyPrice, Qty)

Author-Telephone(Email*, Telephone)

3 Part B: Relational Database Model

1) UPDATE employee SET SSN='666884444' WHERE SSN = '123456789';

UPDATE employee SET SSN='123456789' WHERE SSN = '666884444';

- a) SSN is a primary key in employee table and first and second query try to update the SSN with another already existing value. So, it will **violate the Key Constraint** which states that primary key should be unique.
- b) Another way to correct the error is to take temporary rows to store these data with temporary ssn and then delete the original data and add new rows with data values of temporary rows and original ssn. Make changes to related reference tables parallelly.
- c) Correct List of SQL Statements for proposed solution is as below :

INSERT INTO employee

**Select fName,mInit,lName,'1',bDate,address,sex,salary,dno,super_SSN
from employee
where ssn='123456789';**

INSERT INTO employee

**Select fName,mInit,lName,'2',bDate,address,sex,salary,dno,super_SSN
from employee
where ssn='666884444';**

UPDATE works_on

**SET essn=(case when essn='666884444' then '2' else '1' end)
WHERE essn IN ('666884444','123456789');**

UPDATE dependent

**SET essn=(case when essn='666884444' then '2' else '1' end)
WHERE essn IN ('666884444','123456789');**

Delete from employee where ssn='123456789';

Delete from employee where ssn='666884444';

INSERT INTO employee

Select

fName,mInit,lName,'666884444',bDate,address,sex,salary,dno,super_SS
N from employee
where ssn='1';

INSERT INTO employee

Select

fName,mInit,lName,'123456789',bDate,address,sex,salary,dno,super_SS
N from employee
where ssn='2';

UPDATE works_on

SET essn=(case when essn='1' then '666884444' else '123456789' end)
WHERE essn IN ('1','2');

UPDATE dependent

SET essn=(case when essn='2' then '123456789' else '666884444' end)
WHERE essn IN ('1','2');

Delete from employee where ssn='1';

Delete from employee where ssn='2';

2) UPDATE employee SET salary = salary*110/100 WHERE SSN = '888665555';

➔ No Violations. As we are updating a value of attribute which is neither a primary key nor a foreign key referencing to any primary key of another entity. So it will just update the value of salary attribute without violating any constraints.

3) UPDATE employee SET super_SSN = '666884444' WHERE SSN = '123456789';

➔ No Violations. Because new value of super_SSN column already exists in the SSN column and hence does not violate Referential integrity constraint.

4) INSERT INTO project (pname, plocation) VALUES ('Agent X', 'Washington');

➔ **Violates Entity Integrity Constraint.** Because the value of attribute Pnumber in the insert statement cannot be NULL or BLANK, i.e. it requires a value as it is a

primary key in PROJECT table. Here in the insert statement, Pnumber attribute is not given any value and hence violates the above mentioned constraint.