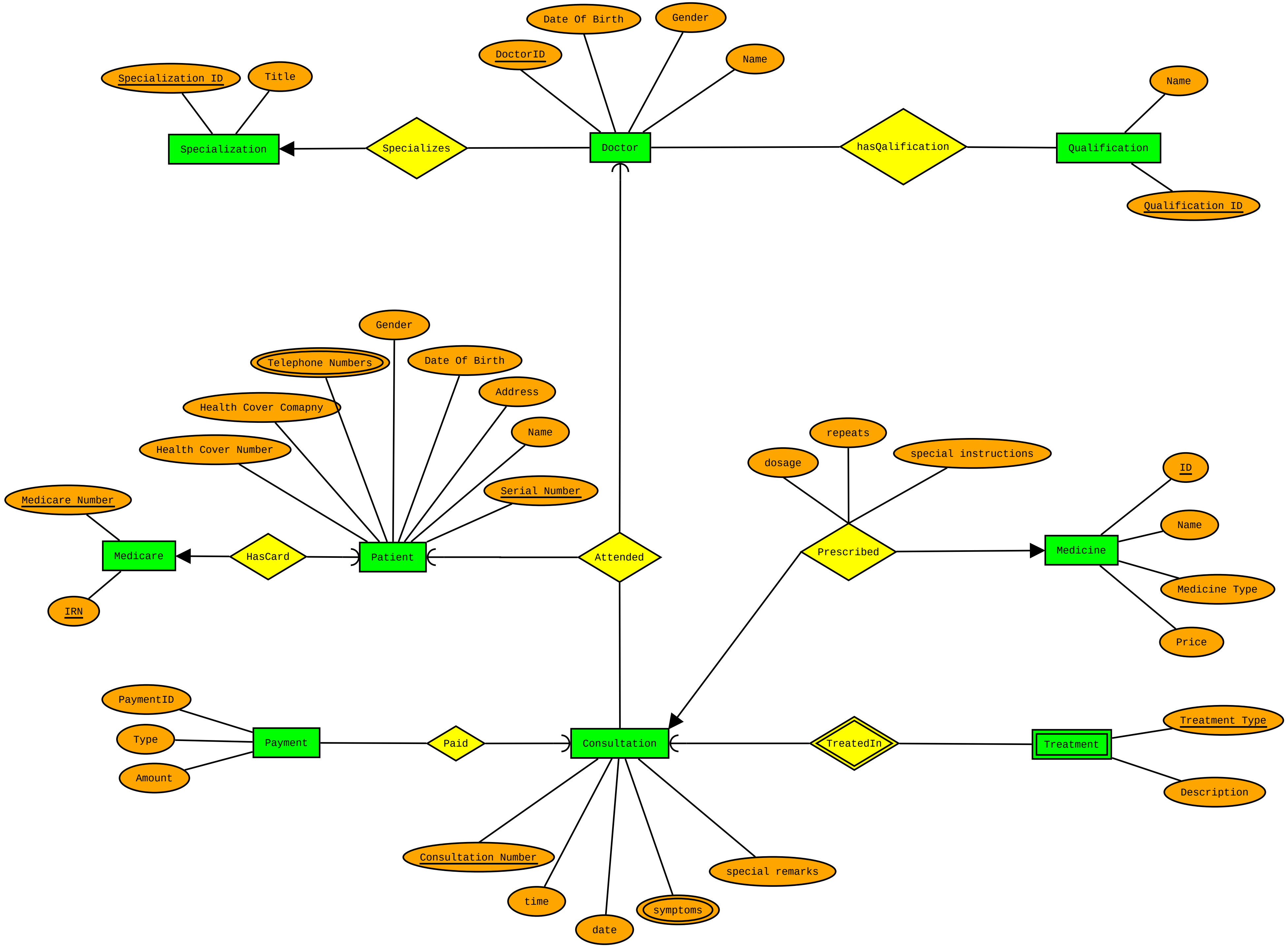
DATABASE CONCEPTS

ASSIGNMENT 1

PAUL KING S3349513

Part A: Entity Relationship Modelling



Part B: SQL

```sql
-- Q1 Display the titles of all books on the subject "DataBases". Your result
-- set should be sorted on the alphabetical order of the titles.
SELECT title
FROM book, subject
WHERE book.subjectID = subject.subjectID
AND subject.subjecttype like '%DataBase%';

-- Q2 Display
  -- 1. the number of books on the subject "DataBases".
SELECT count(title)
FROM book, subject
WHERE book.subjectID = subject.subjectID
AND subject.subjecttype like '%DataBase%';

  -- 2. the number of book copies on the subject "DataBases".
SELECT count(bookID)
FROM book_copy, book, subject
WHERE book_copy.bookdescID = book.bookdescID
AND book.subjectID = subject.subjectID
AND subject.subjecttype like '%DataBase%';

 -- Q3. Display the firstname and lastname of the authors who wrot books on the
 -- subject "Databases".


  -- 1. Write your query without using NATURAL JOINs
SELECT firstname, lastname
FROM author, written_by, book, subject
WHERE author.authorID = written_by.authorID
AND written_by.bookdescID = book.bookdescID
AND book.subjectID = subject.subjectID
AND subject.subjecttype like '%DataBase%';

 -- 2 Write your query using NATURAL JOINs
-- 2 Write your query using NATURAL JOINs
SELECT DISTINCT firstname, lastname
FROM (select * from author
NATURAL JOIN written_by), book, subject
WHERE author.authorID = written_by.authorID
AND written_by.bookdescID = book.bookdescID
AND book.subjectID = subject.subjectID
AND subject.subjecttype like '%DataBase%';

-- Q4. Whor translated the "book "American Electrician's Handbook"? Display the
-- firstname, middlenames, and lastname of the translator.
SELECT firstname, middlename, lastname
FROM author, written_by, book
WHERE book.title = 'American Electrician''s Handbook'
AND book.bookdescID = written_by.bookdescID
AND role = 'Translator'
AND written_by.authorID = author.authorID;
-- This book is not in the database

-- Q5 Display the firstname and lastname of the people who returned books late
SELECT person.firstname, person.lastname
FROM person, borrow
WHERE person.personID = borrow.personID
AND returndate > duedate;
```

```sql
-- Q6 Display the firstname and lastname of the people who returned books more
-- than 7 days late
SELECT person.firstname, person.lastname
FROM person, borrow
WHERE person.personID = borrow.personID
AND returndate > duedate + 7;

-- 7 Display the titles of books that havn't been borrowed
SELECT DISTINCT book.title
FROM book, book_copy
WHERE book.bookdescID = book_copy.bookdescID
AND book_copy.bookID NOT IN (SELECT bookID FROM borrow_copy);

/* 8 A borrower wants to borrow the book titled "PRINCIPLES AND PRACTICE OF
DATABSE SYSTEMS", but all of its copies are already borrowed by others. Write
two queries to display other reccomended titles usging the follwing methods. */
  -- 1. Using partial matchin for the inclusion of the word "DATABASE" in the
  -- book title
SELECT title
FROM book
WHERE title like '%DATABASE%'
AND title != 'PRINCIPLES AND PRACTICE OF DATABASE SYSTEMS';

  --2. By searching of other books written by the same author (i.e. the author
  --of "PRINCIPLES AND PRACTICE OF DATABASE SYSTEMS")
SELECT author.*
FROM author, written_by, book
WHERE author.authorID = written_by.authorID
AND written_by.bookdescID = book.bookdescID
AND book.title = 'PRINCIPLES AND PRACTICE OF DATABASE SYSTEMS';

SELECT book.title
FROM book, written_by, author
WHERE book.bookdescID = written_by.bookdescID
AND written_by.authorID = author.authorID
AND author.authorID = '2492';

/* 9 Display the list of publishers who have published books on the subject
"DataBases". Your query should display publisher's full name, along with book
titles they published. Please note that, these publishers may have published
books in other subjects too. However, your query should only display book titles
in DataBases subject. */
Select publisher.publisherfullname, book.title
From publisher, published_by, subject, book
where publisher.publisherID = published_by.publisherID
AND published_by.bookdescID = book.bookdescID
AND book.subjectId = subject.subjectID
AND subject.subjecttype = 'DataBases';

-- 10 List the full names of publishers who have not published books on the
-- subject "Databases"
SELECT DISTINCT publisher.publisherfullname
FROM publisher, published_by, subject, book
WHERE publisher.publisherID = published_by.publisherID
AND published_by.bookdescID = book.bookdescID
AND book.subjectId = subject.subjectID
AND 'DataBases' NOT IN subject.subjecttype;
```

Part C- Relational Database Model

1. Insert <'Robert', 'F', 'Scott', '987654321', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1 > into EMPLOYEE

No violations.

2. Insert <'ProductA', 3, 'Bellaire', 2> into PROJECT.

Dnum is a foreign key to Dnumber in apartment. The value '2' does not exist in the primary key Dnumber making this '2' invalid as it violates the referencial integrity constraint/foreign key constraint.

3. Insert <'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT.

Violates referential integrity/foreign key constraint

'Mgr_ssn' is a in 'DEPARTMENT' is a foreign key to its parent 'Ssn' in 'EMPLOYEE'. The value '943775543'.

4. Insert <'123454321', null, '40.0'> into WORKS_ON.

Entity Integrity Constraint violated.

Pno in the table WORKS_ON is part of a primary key. The statement is trying to insert a null value into this column and key tuples cannot be null.

5. Insert <'453345453', 'John', M, '12-DEC-60', 'SPOUSE'> into DEPENDENT.

No Violations.

6. Delete the DEPENDENT tuples with dependent_name= 'Joy'.

No Violations

7. Delete the EMPLOYEE tuple with SSN= '888665555'.

No Violations

8. Modify the SUPER_SSN of the EMPLOYEE tuple with SSN='123456789' to '666884444'.

No Violations.