



School of information technology and engineering

Program: B.Tech

Course: CSE2004 - DBMS

Database Management Systems Lab

Cycle Sheet 2

Hospital Database

Doctor (Doc_ID, Doc_Name, Gender, DOB, Specialist, Qualification, Contact, Address, Dept_No)

Department (Dept_No, Dept_Name, Room_No, Floor, HOD, Estd_Date)

Staff (Staff_ID, Staff_Name, Category(nurse, lab technician, cashier, security), Designation, DOB, Contact, Address, Dept_No)

Patient (Pat_ID, Pat_Name, DOB, Gender, Contact, Address)

In_Patient (Pat_ID, Date_Of_Admission, Bed_No, Start_Time, End_Time)

In_Patient_Prescription(Pat_ID, Pres_ID)

Appointment (App_ID, Pat_ID, Doc_ID, Nurse_ID, Consult_Room_No, Date, Time)

Prescription (Pres_ID, App_ID, Date, Time, Diagnosis_Detail)

Prescribed_Medicines (Pres_ID, Medicine_Name, Dosage, Brand)

Hospital_Bill (Inv_No, Inv_Date, Pat_ID, Bill_Amount, Payment_Type (cash/credit card/debit card), discount)

Lab_Tests (Test_ID, Pat_ID, Date, Time)

Test_Results (Test_ID, TT_ID, Result)

Test_Types (TT_ID, Description, Low_Value, High_Value, Test_Method, Technician)

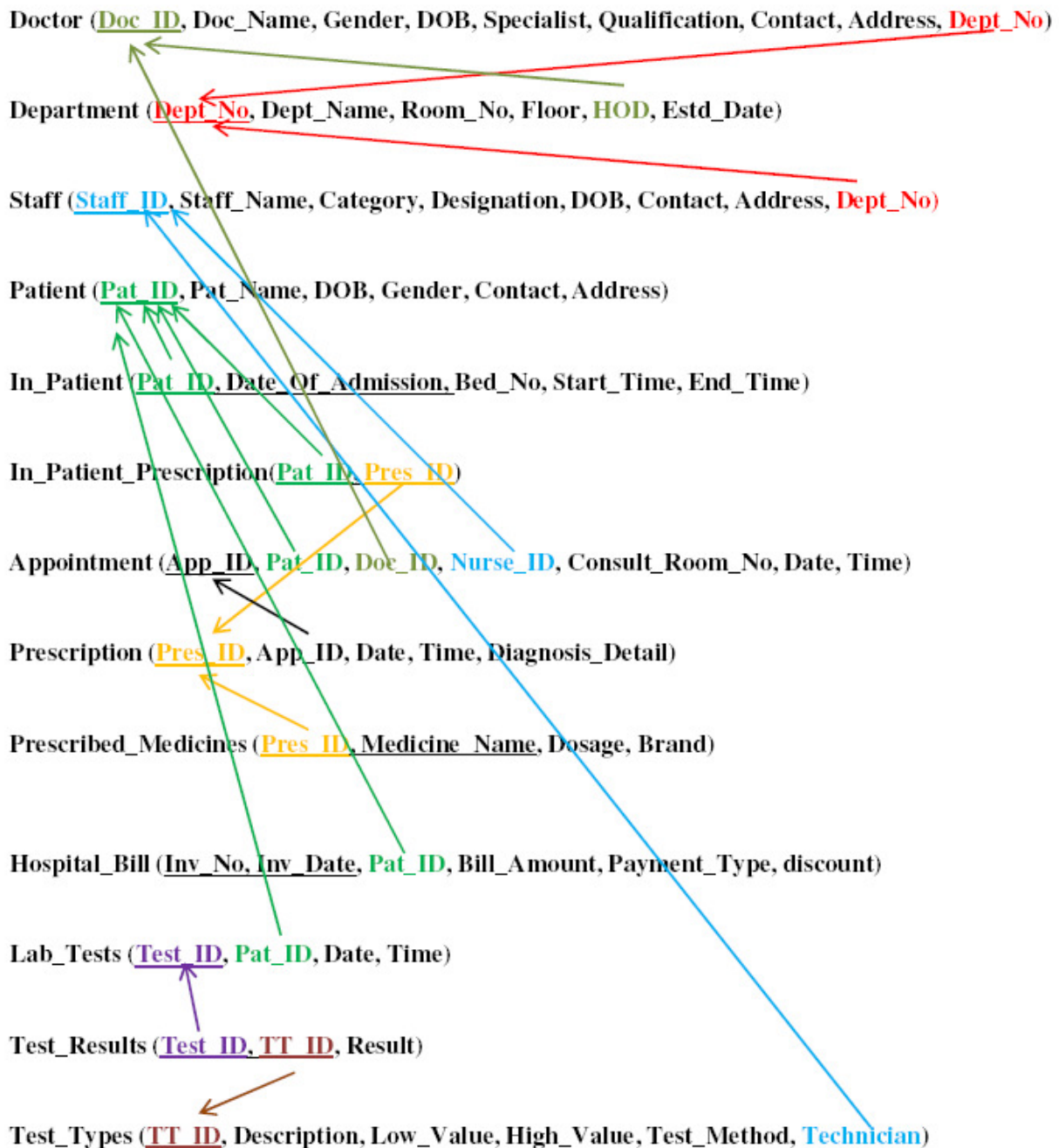


Figure 1: Primary key and foreign keys

Questions:

DDL statements (ALTER, CONSTRAINT etc):

1. Modify Hospital_Bill by adding an attribute consulting_physician and add foreign key constraint for that attribute. Use constraint name for foreign key constraint.
2. In Patient table, replace address with three attributes namely street, city and pincode.
3. Add an attribute Test_Time which can accept only two values “Before food” and “After food” with proper constraint name.
4. Remove the constraint only from test_time attribute.
5. Drop address attribute from staff table and add attributes door_no, street, city, and pincode.
6. Create a table Medicines with schema medicines=(med_name, brand, dosage, manu_date, exp_date). Ensure that manu_date should not be later than exp_date. Create an appropriate constraint to ensure this.
7. Remove the attributes dosage and brand from Prescribed_Medicines and alter the medicine_name attribute as a foreign key referencing the new table Medicines.
8. Create a view for doctors who are specialized in ‘Cardiology’ from Doctor table with attributes doc_id, doc_name and gender.
9. Add an attribute No_of_staff in Department table and create a constraint with constraint name to make sure the number is >0.
10. Add an attribute with In_Patient_prescription to store the Room_Type which can store the values “AC” and “Non-AC”.

SQL queries with JOIN operation (Practice these queries with Cartesian product, inner join, natural join etc.)

1. Find the HOD of doctor ‘Raghavan’ (Hint: you need to join the tables DOCTOR and DEPARTMENT)
2. Find the list of all patients who were admitted in bed number ‘B101’
3. Display all the prescribed medicines of patient with Pat_ID ‘P101’
4. Display the test results of patient ‘Mani’
5. Display all bills of bill amount more than 10000 rupees and paid by the patient ‘Steve’.

6. Find the category and address of the nurse who attended the patient with pat_no 'P220'.
7. Find the list of doctors who worked in the department which is started on or after '10-May-2018'.
8. Get the name of technicians who tests blood glucose level.
9. Display the details of all patients who were hospitalized between '10-Mar-2017' and '10-Apr-2017'
10. Display the in-patient prescription of the patient whose name is 'Gayle'.

SQL queries with AGGREGATE and CHAR functions

1. Find the number of doctors who are working in the department 'D101'
2. Count the number of male patients who are treated by the doctor with doctor id 'D101'
3. Find the total bill paid by the patient 'Karthik'
4. Find the name and address of the patient who paid the highest bill of all patients.
5. Get the specialization of doctors whose name start with the letter 'M'
6. Find the all the patients details whose name is exactly 5 characters long
7. Display the department names in ascending order
8. Get the gender wise count of patients.
9. Get the count of doctors for each specialization.
10. Get the total number tests conducted in any particular date.

SQL queries - Nested subqueries

1. All of the queries in "SQL queries with JOIN operation" section can be tried with subqueries concept.
2. Display the list of doctors who are working in the department with more number of doctors using sub-query and IN operator.
3. Find the name and id of all patients who are older than all the doctors in the entire 'cardiology' department. Use subqueries and ALL operator.

4. Find the prescription ids of all prescription that included a medicine from the brand 'Ranbaxy' using nested subqueries.
5. Find the list of patients who paid their bill through either 'credit card' or 'debit card' using subquery.

SQL queries using other functions

Practice queries using DATE, NUMERIC, and CHARACTER functions. Refer DBMS_Lab_Reference_Material.pdf file. Try to upload at least two queries from each function category.

Example: Display the doctor names in capital letters. `SELECT UPPER(Doc_Name) FROM Doctor;`
