Hospital Admission Management System

## Overview

To design the database, insert some sample data, and implement a set of required features. Each feature will be implemented as one or more Oracle PL/SQL procedures/functions. You do **NOT** need to write a graphic user interface.

**Assumptions:**

1. The system will store information about hospitals, including hospital ID, name, address.
2. The system will store information about doctors, including doctor ID and name. A doctor is affiliated with one or more hospitals (affiliated means the doctor can treat a patient such as performing a surgery at that hospital).
3. The system will store information about nurses. Each nurse has an ID, name, and the hospital the nurse is working for.
4. The system stores information about rooms. Each room has a unique room ID, the hospital the room belongs to, a detailed location (e.g., Building A room B), and a room type. There are three types of rooms in the system: regular room (where a patient can stay and be monitored), operating room (where a surgery can take place), and ICU (intensive care unit).
5. The system stores shift information for nurses, including a unique shift ID, nurse ID, room id, shift start and end time. A shift is usually 8 to 12 hours long.
6. During a shift a nurse is assigned to one or more rooms.
7. The system stores patient information, including a unique patient ID, name, gender, date of birth, address, phone, and email.
8. The system stores information about a patient's admission into a hospital, including a unique admission ID, patient ID, hospital ID, doctor ID, admit date, discharge date, reason for admission, notes for discharge (e.g., a textual summary of what treatment the patient received etc. ), and a status column. The status can be 1:'admitted waiting for a room', 2:'room assigned', 3:'discharged'
9. A patient during an admission can be assigned to one room or different rooms during different time. For example, for a surgery the patient is usually first assigned to a regular room to wait for the surgery to start, then be assigned to an operating room to have the surgery, and then to another regular room to be monitored while recovering from anesthesia.
10. The system stores a list of treatment (like different types of surgeries, intensive care, etc.). Each treatment has an ID and description of what the treatment is (e.g., ankle surgery, full body anesthesia, intensive care, x-ray).
11. The system stores the treatments a patient received in an admission, including date of the treatment, treatment ID, admission ID).
12. The system stores commonly used drugs, including drug ID, name, dose\_value (e.g., 500 mg a time), doses per day (e.g., 3 times per day).
13. The system stores drug allergy of patients, including patient ID and ID of drug the patient is allergic to.
14. The system stores the prescription given in an admission, including a unique prescription ID, drug ID, admission ID, prescription date, number of days to take this drug, number of refills.
15. The systems stores a message table for doctors or patients. The message table has a unique message ID, doctor ID (for which doctor this message is for), patient ID (for which patient this message is about), message time, and message body.

**Feature 1:** add a patient. Input: name of patient, gender, date of birth, address, phone and email. This procedure does the following:

1. First check whether a patient with the same name and date of birth exists. If so print out a message that the patient already exists and update address, phone, and email.
2. If the patient does not exist, generate a new patient ID and insert a new row into patient table with the input information and print out newly assigned patient ID.

**Feature 2:** admit a patient. Input includes name of a patient , date of birth, hospital ID, doctor ID, reason for admission. This procedure does the following

1) First checks whether there is a patient with the given name and date of birth. If there is no match print an error message and stop.

2) If there is a match, check whether the input hospital ID is valid. If it is not valid, print an error message and stop.

3) check whether there is a doctor with the given ID and affiliated with the hospital. If this is not true, print an error message and stop.

4) If all checks in step 1 to 3 are fine, insert a row into admission table with a newly assigned admission ID and given patient ID, hospital ID, doctor ID, and date as current date. You can leave discharge date and discharge note as NULL, and status as admitted waiting for a room.

5) Please also insert a row into message table with given doctor id and patient ID, time

as current time, and body as 'Dr. Y: Patient X has been admitted into hospital Z', where Y is name of doctor, X is name of patient, Z is name of hospital.

**Member 2:**

**Feature 3\*\*\*:** Assign a room to a patient for a given admission.

Input: ID of admission, a start time of using the room, an end time, and a room type. This procedure does the following.

1. Check whether the admission ID is valid. If not print an error message and stop
2. Check in the hospital the admission belongs to, whether there is a room with the input type that is not assigned for a time that overlaps with the start and end time.

Hint: two time intervals overlap if start time of interval 1 <= end time of interval 2 and end time of interval 1 >= start time of interval 2.

1. If there are at least one such room, print out the room ID and location, and assign the first available room to the admission, and insert a message to the message table with doctor ID and patient ID as the corresponding doctor ID and patient ID for that admission, and the message body saying 'Room X has been assigned to Patient Y from Z to W', where X is room's location, Y is name of patient, Z is start time and W is expected end time.
2. Update status of the admission to room assigned.
3. If there is no room available, print out a message no rooms are available.

**Member 3:**

**Feature 4:** Add a treatment to a patient.

Input: an admission ID, a treatment ID and a treatment date.

1) This procedure first checks whether the treatment ID is valid. If it is not, print an error message and stop.

2) It then checks whether the admission ID is valid. If it is not, print an error message and stop.

3) It then inserts a row into admission\_treatment table, with given admission ID, treatment ID and date. Please insert a message into message table with patient ID and doctor ID as corresponding patient ID and doctor ID for that admission, and body as 'Patient X received treatment Y on Z' where X is name of patient, Y is description of treatment, and Z is date of treatment.

**Feature 5:** Add a list of drug allergy for a patient.

Input: patient ID, and an array of drug IDs that the patient is allergic to.

1) This procedure will first check whether the patient ID is valid. If it is not valid, print out an error message and stop.

2) It then loops through each drug ID in the array, and checks whether each drug ID is valid, if it is not valid, print a message the drug ID is not valid and skip that drug ID.

3) If the drug ID is valid, it will check whether the pair of patient ID, drug ID is already in the drug\_allergy table. If so print a message 'already inserted' and skip this drug ID.

4) If this pair is not in drug\_allergy table, insert this pair and print 'Allergy to drug X recorded', where X is name of drug.

**Member 4:**

**Feature 6\*\*\*:** Prescribe a drug.

Input includes: a drug ID, admission ID, date of prescription, #of days to take the drug, #of refills. This procedure does the following

1) the procedure first checks whether the drug ID is valid. If it is not, print an error message and stop.

2) It then checks whether the admission ID is valid, if not, print an error message and stop.

3) It then checks whether the corresponding patient is allergic to this drug. If so, print a message 'Patient X is allergic to Y, choose another drug', where X is name of patient, Y is name of drug. It then stops.

4) If there is no allergy, insert a row into prescription table with an automatically generated prescription ID, given drug ID, admission ID, prescription date, #of days and #of refills.

5) Insert a message into message table with appropriate patient and doctor ID, and the message body says 'A new prescription of Y is created', where Y is name of drug.

**Member 4:**

**Feature 7:** show rooms and patients a nurse is assigned to. Input includes a nurse ID and a specific time.

1) This procedure first checks whether the nurse ID is valid. If not, print out an error message and stop.

2) It next checks whether the nurse is on a shift at the input time. If not, print the nurse is not working at that time and stop.

3) If the nurse is working on a shift at the input time, print out for each room the nurse is assigned to, the room location, and name of patients who is assigned to the room at that time.

**Feature 8:** Discharge a patient.

Input includes admission ID, discharge date, discharge note. It does the following.

1) it first checks whether the admission ID is valid. If not, print an error message and stop.

2) it then updates admission table to set discharge date and discharge note, and change status to discharged.

3) it inserts into message table a row with corresponding patient ID and doctor ID, with the message body 'Patient discharged on X' where X is discharge date.

4) it prints out a list of treatments received during this admission, including description of treatment and date of treatment.

5) it prints out all prescriptions for this admission, including name of drug, dose value, dose per day, number of days, and number of refill.

**Group features:** ALL group features need to be implemented. Group features will be graded group-wise.

**Feature 9\*\*\*:** Contact tracing. Input: name of a patient, DOB, and a date that the patient is in hospital. The procedure does the following:

1) check whether there is such a patient (with given name and DOB). If not, print an error message and stop.

2) If there is such a patient, check if the patient was admitted in the hospital on the given date (i.e., between admit\_date and discharge\_date). If the patient is not admitted on that date, print a message 'This patient is not in hospital that day' and stop.

3) If the patient is in hospital that day, print out name of the doctor who oversees the patient's admission, all rooms the patient was assigned to and start and end time of each assignment.

4) For each room assignment in step 3, print out names of nurses as well as their shifts' start and end time for those who were assigned to the room with duration overlaps with duration of the patient was assigned to that room.

Again two duration overlap if

start time of duration 1 <= end time of duration 2 and end time of duration 1 >= start time of duration 2.

**Feature 10\*\*\*:** Room statistics. Input: a start date (v\_start\_date) and an end date (v\_end\_date), and a hosptial ID. This procedure does the following:

1) first checks whether the hospital ID is valid. If it is not, print out an error message and stop.

2) For each room in the hospital and each date between the start and end date (including both), check whether the room is assigned to at least one patient during that day.

This means that there exists an assignment whose start and end time overlaps with the start and end time of that day (you can consider a day starts from 12 am and ends at 11:59 pm).

3) Add up the #of days for each room that there is a patient assigned (the condition checked in step 2)

4) Divide the number of days a room is occupied by in step 3 by total number of days between input start and end date. To compute number of days, you can just subtract one date from the other.

5) Print room location, room type, and occupancy rate computed in step 4) for each room in that hospital.

**Feature 11\*\*\*:** Compute shift statistics. Input is a hospital ID and start and end date. You can assume that the gap between end and start date is longer than any shift. This procedure does the following.

1) The procedure first checks whether the hospital ID is valid. If the ID is invalid it prints an error message and stops.

2) It then computes for each nurse working at the hospital, the number of hours worked between 8 am on the start and 8 am on the end date. You need to consider the case a shift starts before 8 am on the start date or ends after 8 am on the end date. In such cases, only count number of hours inside the given time period (8 am of start date to 8 am of end date).

3) It then counts the number of days from the start date to the end date.

4) It computes average number of hours per day the nurse is working, which is the total number of hours / number of days. Print out name of nurse and average number of hours.

5) it finally computes an average number of hours work per day for all nurses in the hospital, which is the average of every nurse's average daily work hours.

Hint:

1) to convert a variable x which is interval day to second type into hours.

You can do

extract(day from x)\*24 + extract(hour from x) + extract(minute from x)/60

The idea is to get #of days from the interval and multiply by 24 (hours per day), then extract number of hours, and then extract number of minutes and divide by 60 (minutes per hour). The sum is hours (with decimal points).

2) Oracle does not support sum over interval type, but you can add to interval type using +.

So in step 2) you can write a cursor to get start and end time of each nurse shift fall between input start and end time, and add the duration of each shift (as interval) up in a loop. Another option is to convert the interval to number of hours as hint 1 and then sum them up.

**Feature 12\*\*\*:** Analyze admission/re-admission. This procedure takes an interval as input. It does the following.

1) It lists all names of patients with pair of admissions and the gap between previous admission's discharge date and next admission's admit date is less than the input interval.

2) For each such pair of admissions found in step 1), print out discharge date of first admission and admit date of second admission and reasons for each admission.

3) For each hospital, compute readmission rate as fraction of patients identified in step 1). E.g., if a hospital admitted 5 patients, but 2 of them have readmission defined in step 1), the rate is 0.4. Print out name of hospital and readmission rate.