# Assignment 3 5%

## Due Date: November 23, before midnight upload to BLearn

**Study material:** Chapters 12, 13, 14.Read Appendix D for the summary of database design.

**Part 1 3 %**

j0286864%5b1%5d**Wellmeadows Hospital Case Study**

This assignment is based on the **Wellmeadows Hospital Case Study** from the textbook’s **Appendix B.3.** Study the data and transaction requirements and prepare the following documentation. You may have to make some assumptions about the hospitals in general (based on your experience or research on the Web) or talk to some experts from the medical field. Please document your assumptions.

The **scope of the assignment** is limited to:

* **Patient data** (see patent registration form, Fig. B.3). your model should include all required data to create the Wellmeadows Hospital registration form (including the next-of-kin details and the local doctor details).
* **Patient medication** (see Fig. B.5) and pharmaceutical supplies. Note: The hospital decided to maintain data on medication prescribed to all patients (inpatients and outpatients) and the patient medication form does not require the ward number and name (the patient/bed allocation is excluded from our current scope).
* The results from the **STOPBang** questionnaire for the patients (an additional requirement not included in the Appendix Case Study). Add a calculated attribute: risks for obstructive sleep apnea OSA.

1. **Conceptual database model:** Create a conceptual data model using ER modeling technique (Chapter 12). Define naming conventions for the entities, relationships, and attributes and use these standards in your design. If your model requires additional concepts such as specialization/generalization, aggregation, or composition, you may use EER modeling techniques (Chapter 13). Using PowerDesigner as a CASE tool, create a Conceptual Data Model (CDM) for the system**. Create an ERD diagram for the data model including entities, attributes, primary keys, conceptual data types, relationships and their cardinalities.** Export the graph to a Word Document. Make sure that your graph shows all the required components (check the preferences). The graph will be marked for the readability and presentation.

****

**CONCEPTUAL DATA MODEL**

****

**PHYSICAL DATA MODEL**

1. **Physical database model:** Using your CDM, generate physical model for Oracle 10g (depending on your assumptions and design, introduce necessary changes). Generate the DDL script and **implement the model using TRU server for Oracle 12c. Add two (or more) secondary indexes and describe how adding indexes will improve the performance of the system.** Create **only** the tables required for the scope of the assignment. Write SQL query to list **the tables** and **indexes** created for Assignment 3 (the objects should have names with prefix A3). List the object name, creation date (using ISO standard). Hint: use USER\_OBJECTS view of data dictionary in Oracle.

CREATE INDEX a3\_p\_m\_secondary\_index ON

AS3\_PRESCRIPTIONS (prescriptionid,

patient\_number)

COMPUTE STATISTICS;

CREATE INDEX A3\_PATIENT\_SECONDARY\_INDEX

ON AS3\_PATIENT (

patient\_number,

ld\_id,

patient\_first\_name,

patient\_last\_name)

COMPUTE STATISTICS;

SELECT OBJECT\_NAME, to\_char(CREATED,'YYYY-MM-DD HH:mi:ss'),OBJECT\_TYPE

FROM USER\_OBJECTS

WHERE

UPPER (OBJECT\_TYPE) = UPPER('TABLE')

AND EXTRACT (YEAR FROM CREATED) = EXTRACT (YEAR FROM SYSDATE)

AND EXTRACT (MONTH FROM CREATED) = EXTRACT (MONTH FROM SYSDATE)

AND EXTRACT (DAY FROM CREATED) = EXTRACT (DAY FROM SYSDATE)

UNION

SELECT OBJECT\_NAME, to\_char(CREATED,'YYYY-MM-DD HH:mi:ss'),OBJECT\_TYPE

FROM USER\_OBJECTS

WHERE

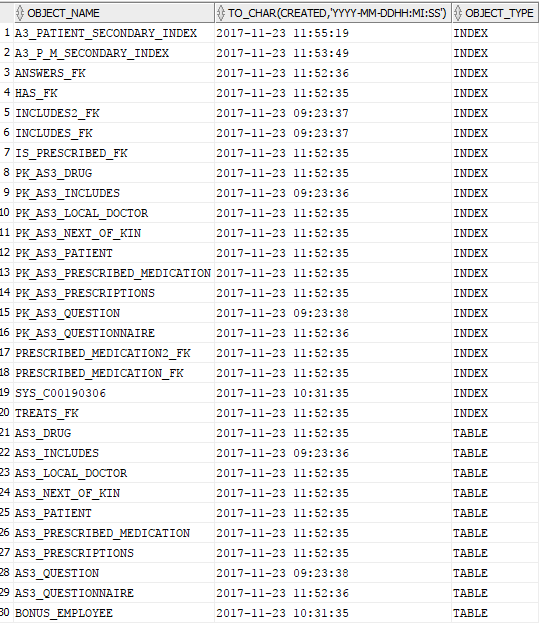
UPPER (OBJECT\_TYPE) = UPPER ('INDEX')

AND EXTRACT (YEAR FROM CREATED) = EXTRACT (YEAR FROM SYSDATE)

AND EXTRACT (MONTH FROM CREATED) = EXTRACT (MONTH FROM SYSDATE)

AND EXTRACT (DAY FROM CREATED) = EXTRACT (DAY FROM SYSDATE)

ORDER BY OBJECT\_TYPE;



**Part 2 2%**

1. Write a **trigger** to add automatically a new patient number using the format from the Form on page 390, i.e., letter “P” followed by 5 digits. Use a sequence to generate numbers, and start the sequence from 10034, incrementing by 1, maximum 99999, no cache. Print the create statement for the trigger and test results to demonstrate that the trigger is working.

CREATE SEQUENCE A3\_PATIENT\_NUMBER\_SEQ

START WITH 10034

INCREMENT BY 1

MAXVALUE 99999

NOCACHE;

CREATE OR REPLACE TRIGGER AS3\_PATIENT\_NUMBER\_TRIGGER

BEFORE INSERT ON AS3\_PATIENT

FOR EACH ROW

BEGIN

:new.patient\_number :='P'||(A3\_PATIENT\_NUMBER\_SEQ.NEXTVAL);

END;

1. **Add test data to the tables.** Include the data from the sample form (B.5 Patient Medication Form). Write an SQL query to list all data (information) required by the Patient Medication Form (excluding the ward/bed data). Print the query and the results.

SELECT

P.patient\_first\_name "Patient First Name",

P.patient\_last\_name "Patient Last Name",

P.patient\_dob "Patient DOB",

P.phone\_number "Patient Phone",

P.gender "Patient Gender",

P.registration\_date "Patient Registration Date",

P.patient\_address "Patient Address",

"LD".LD\_FULL\_NAME"LD Full Name",

"LD".LD\_CLINIC\_ADDRESS,

"LD".LD\_TELEPHONE\_NUMBER,

"LD".LD\_CLINIC\_NUMBER,

"NOK".NOK\_FULLNAME,

"NOK".NOK\_ADDRESS,

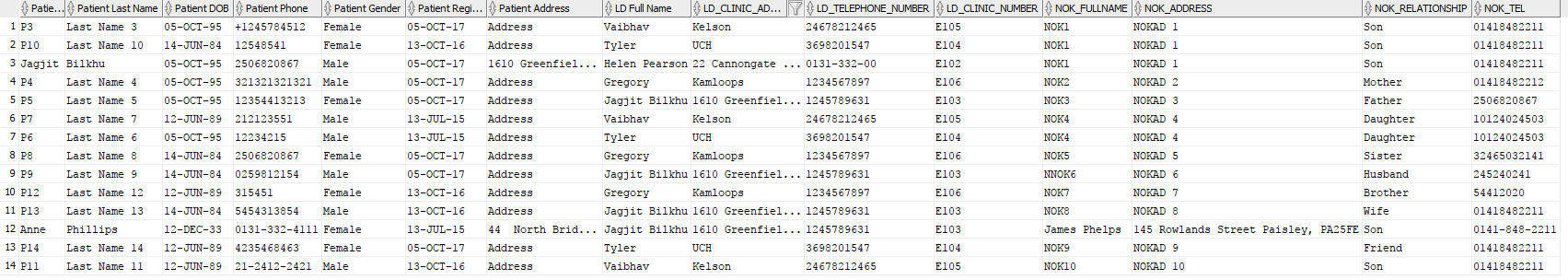
"NOK".NOK\_RELATIONSHIP,

"NOK".NOK\_TEL

FROM

as3\_patient P JOIN AS3\_NEXT\_OF\_KIN "NOK" ON P.NOK\_ID = NOK.NOK\_ID

JOIN AS3\_LOCAL\_DOCTOR "LD" ON P.LD\_ID = LD.LD\_ID;



1. **Create a PL/SQL procedure to list the patient’s details for a specified Patient Number (as an input parameter).** The stored procedure should list all the data required for the patient registration form (fig. B.3). the dates should be printed using ISO standard. The layout of the data can be simplified, but the captions and data should be readable.

CREATE OR REPLACE PROCEDURE GET\_PATIENT\_DATA(p\_patient VARCHAR2)

AS

v\_patient AS3\_PATIENT%ROWTYPE;

BEGIN

SELECT \* INTO v\_patient FROM as3\_patient WHERE UPPER(p\_patient) = UPPER(patient\_number);

DBMS\_OUTPUT.PUT\_LINE('First Name: '||v\_patient.patient\_first\_name||chr(10)

||'Last Name: '||v\_patient.patient\_last\_name||chr(10)

||'DOB: '||to\_char(v\_patient.patient\_dob,'DD/MM/YYYY')||chr(10)

||'Phone: '||v\_patient.phone\_number||chr(10)

||'Gender: '||v\_patient.gender||chr(10)

||'Registered Date: '||to\_char(v\_patient.registration\_date,'DD/MM/YYYY')||chr(10)

||'Address: '||v\_patient.patient\_address);

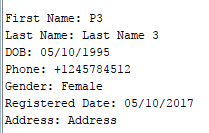
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('----------NO PATIENT WITH THAT ID FOUND----------');

END;

EXECUTE GET\_PATIENT\_DATA('P10037');



1. **Create a PL/SQL procedure to list the patients** who have high and very high risks for OSA (assessed by STOPBang) and who are **currently (today) taking morphine** (use SYSDATE as today’s date). Print the patients’ ids, names, STOPBang scores, risks, and information about the intake of morphine (dosage, method, units per day, start, and end date). Print the stored procedure and the results from test.

--This it to keep track when checking the data conditions for OSA risk in the procedure

ALTER TABLE AS3\_QUESTIONNAIRE

ADD (total\_sum NUMBER(5) GENERATED ALWAYS AS (snoring+tired+observed+pressure+bmigreaterthan35+ageolderthan50+largeneck+gendermale) VIRTUAL,

stop\_sum NUMBER(5) GENERATED ALWAYS AS (snoring+tired+observed+pressure)VIRTUAL);

create or replace PROCEDURE GET\_PATIENT\_RISKS

IS

vpatient VARCHAR2(10);

vb NUMBER(1);

vg NUMBER(1);

v\_sum NUMBER(5);

v\_stop\_sum NUMBER(5);

v\_risk VARCHAR2(25);

v\_fname VARCHAR2(15);

v\_lname VARCHAR2(15);

v\_start DATE;

v\_end DATE;

v\_units NUMBER(10);

v\_admin\_method VARCHAR2(60);

v\_drugName VARCHAR2(15);

CURSOR assesmentCursor IS

SELECT P.PATIENT\_NUMBER,P.PATIENT\_FIRST\_NAME,P.PATIENT\_LAST\_NAME, M.START\_DATE, M.END\_DATE,

M.UNITS\_PER\_DAY, D.METHOD\_OF\_ADMINISTRATION,D.DRUG\_NAME,Q.BMIGREATERTHAN35,Q.TOTAL\_SUM,Q.STOP\_SUM,Q.GENDERMALE

FROM AS3\_QUESTIONNAIRE Q

JOIN AS3\_PATIENT P

ON Q.PATIENT\_NUMBER = P.PATIENT\_NUMBER

JOIN AS3\_PRESCRIPTIONS A ON A.PATIENT\_NUMBER = P.PATIENT\_NUMBER

JOIN AS3\_PRESCRIBED\_MEDICATION M ON M.PRESCRIPTIONID = A.PRESCRIPTIONID

JOIN AS3\_DRUG D ON M.DRUG\_ID = D.DRUG\_ID

GROUP BY P.PATIENT\_NUMBER,P.PATIENT\_FIRST\_NAME,P.PATIENT\_LAST\_NAME, M.START\_DATE, M.END\_DATE,

M.UNITS\_PER\_DAY, D.METHOD\_OF\_ADMINISTRATION,D.DRUG\_NAME,Q.BMIGREATERTHAN35,Q.TOTAL\_SUM,Q.STOP\_SUM,Q.GENDERMALE;

v\_cursor\_row assesmentCursor%ROWTYPE;

BEGIN

OPEN assesmentCursor;

LOOP

EXIT WHEN assesmentCursor%NOTFOUND;

FETCH assesmentCursor INTO v\_cursor\_row;

vb := v\_cursor\_row.BMIGREATERTHAN35;

v\_stop\_sum := v\_cursor\_row.STOP\_SUM;

v\_sum := v\_cursor\_row.TOTAL\_SUM;

vg := v\_cursor\_row.GENDERMALE;

vpatient := v\_cursor\_row.PATIENT\_NUMBER;

v\_fname := v\_cursor\_row.PATIENT\_FIRST\_NAME;

v\_lname := v\_cursor\_row.PATIENT\_LAST\_NAME;

v\_start := v\_cursor\_row.START\_DATE;

v\_end := v\_cursor\_row.END\_DATE;

v\_units := v\_cursor\_row.UNITS\_PER\_DAY;

v\_admin\_method := v\_cursor\_row.METHOD\_OF\_ADMINISTRATION;

v\_drugName := v\_cursor\_row.DRUG\_NAME;

IF vb=1

THEN

IF v\_sum<4

THEN

v\_risk:= 'OBESE LOW RISK ';

ELSE IF v\_sum<6

THEN

v\_risk:= 'OBESE HIGH RISK ';

ELSE IF v\_sum<9

THEN

v\_risk:= 'OBESE VERY HIGH RISK ';

END IF;

END IF;

END IF;

ELSE IF v\_sum<3

THEN

v\_risk:= 'OBESE LOW RISK ';

ELSE IF v\_sum<5

THEN

v\_risk:= 'OBESE HIGH RISK ';

ELSE IF v\_sum<9 OR v\_stop\_sum=2 AND vg = 1

THEN

v\_risk:= 'OBESE VERY HIGH RISK ';

END IF;

END IF;

END IF;

END IF;

DBMS\_OUTPUT.PUT\_LINE('-------------'||vpatient||'-------------'||chr(10)

||'RISK: '||v\_risk||chr(10)

||v\_fname||chr(10)

||v\_lname||chr(10)

||v\_start||chr(10)

||v\_end||chr(10)

||v\_units||chr(10)

||v\_admin\_method||chr(10)

||v\_drugName);

END LOOP;

CLOSE assesmentCursor;

END;

OUT PUT:

-------------P10034-------------

RISK: OBESE LOW RISK

Jagjit

Bilkhu

23-NOV-17

30-NOV-17

10

Oral

Morphine

-------------P10034-------------

RISK: OBESE LOW RISK

Jagjit

Bilkhu

23-NOV-17

01-NOV-18

50

Oral

Morphine

-------------P10035-------------

RISK: OBESE HIGH RISK

Anne

Phillips

23-NOV-17

01-NOV-18

50

Oral

Morphine

-------------P10035-------------

RISK: OBESE HIGH RISK

Anne

Phillips

23-NOV-17

30-NOV-17

10

Oral

Morphine

-------------P10034-------------

RISK: OBESE LOW RISK

Jagjit

Bilkhu

25-NOV-17

30-NOV-18

10

IV

Tetracycline

-------------P10035-------------

RISK: OBESE HIGH RISK

Anne

Phillips

25-NOV-17

30-NOV-18

10

IV

Tetracycline

-------------P10035-------------

RISK: OBESE HIGH RISK

Anne

Phillips

25-NOV-17

30-NOV-18

10

IV

Tetracycline

**Bonus: 1%**

**Design a drug interaction checker.** Redesign the database to add/change tables to add information about **interactions between drugs**. Create a trigger that checks if the new prescription will interact with the current medication(s) prescribed for the patient. The trigger should add an alert flag to the prescription. Add data to the prescription table. Print the create statement for the trigger and test results to demonstrate that the trigger is working.





**Hand-ins:**

Please hand in this assignment as an upload to Blackboard. Indicate the Part and Problem number for your answers. Your document should have a title page with your name(s), student number(s), course title, and date.

1. Title page with name(s) of student(s)
2. Printouts of answers for part 1 and 2 (and bonus)