# Assignment 1 5%

Introduction to Oracle 12c SQL and DB concepts

## Answers Due Date: October 5, 2017 midnight extended October 6 midnight

**Part 1** Maintaining and redesigning an existing database (2 %)

Use your database created from the SQL Developer Tutorial Script (LAB 1). You must use your server account on BTACS database and perform all queries using your own account. This data base should have three tables.

books ( book\_id, title, author\_last\_name, author\_first\_name, rating)

patrons (patron\_id, last\_name, first\_name, street\_address, city\_state\_zip, location)

transactions (transaction\_id, patron\_id, book\_id, transaction\_date, transaction\_type)

Use the existing database to do the following:

1. **Adding new data (DML):**
   1. Add a new book to the BOOKS table: add **your favorite book** (add a high rating ☺).

INSERT INTO L1\_BOOKS

(

BOOK\_ID,

TITLE,

AUTHOR\_LAST\_NAME,

AUTHOR\_FIRST\_NAME,

RATING

)

VALUES

( 'Z0001', 'You are not A Gadget', 'Lanier', 'Jaron', 10 );

* 1. Borrow your favorite book. Use yourself as a patron (added in Lab1). Return your favorite book.

--To borrow transaction code = 1

INSERT INTO TRANSACTIONS (PATRON\_ID, BOOK\_ID, TRANSACTION\_DATE, TRANSACTION\_TYPE)

VALUES

((SELECT PATRON\_ID FROM patrons WHERE first\_name = 'Mila'), 'ZCCCC', SYSDATE, 1);

--To return transaction code = 2

INSERT INTO TRANSACTIONS (PATRON\_ID, BOOK\_ID, TRANSACTION\_DATE, TRANSACTION\_TYPE)

VALUES

((SELECT PATRON\_ID FROM patrons WHERE first\_name = 'Mila'), 'ZCCCC', SYSDATE, 2);

* 1. List **the last transaction** (all columns) for **your favorite book.**

Print the SQL queries for a-c and the result for c.

-- using subquery to find last transaction

SELECT \*

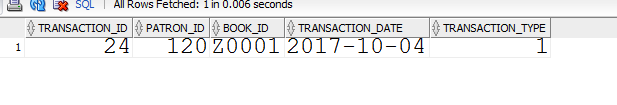
FROM L1\_TRANSACTIONS

WHERE TRANSACTION\_DATE = (SELECT MAX(TRANSACTION\_DATE)

FROM L1\_TRANSACTIONS

WHERE BOOK\_ID ='Z0001')

AND BOOK\_ID = 'Z0001';



1. **Adding new columns (DDL):**
   1. Write an SQL statement to add a new column DOB to the PATRONS table. Print the query.

ALTER TABLE L1\_PATRONS

ADD (DOB DATE);

* 1. Write an SQL statement to add LAST\_MODIFIED and MODIFIED\_BY columns to the PATRONS table. The LAST\_MODIFIED column will have the server date and time and the MODIFIED\_BY will have the USER name.

ALTER TABLE L1\_PATRONS

ADD (LAST\_MODIFIED\_DATE DATE DEFAULT SYSDATE,

MODIFIED\_BY VARCHAR2(15) DEFAULT USER);

Print the DDL statements for a and b.

1. **Changing data (DML):**
   1. Add DOB to an existing patron (yourself). Use the TO\_DATE function for the DOB.

-- please note the my patron id is 120

UPDATE L1\_patrons

SET DOB = to\_date('1999-12-20', 'YYYY-MM-DD')

WHERE PATRON\_ID = 120;

* 1. Add another patron with all data (including columns added in 2b).

-- please note TO\_DATE also DATE function

INSERT INTO L1\_PATRONS

(

PATRON\_ID,

LAST\_NAME,

FIRST\_NAME,

STREET\_ADDRESS,

CITY\_STATE\_ZIP,

DOB

)

VALUES

(

PATRON\_ID\_SEQ.nextval,

'John',

'Smith',

'TRU Way',

'Kamloops',

TO\_DATE('1977-10-09', 'YYYY-MM-DD') );

-- please note location could be null, and the last modified date and modified by have default values

* 1. List the patrons (Last\_modified, patron\_id, last\_name, first\_name, DOB) whose data were added/modified **within last 10 days** (use SYSDATE as today’s date). Use ISO standard for LAST\_MODIFIED including local time. Use only the year for the DOB (YYYY). Sort the results by transaction date (newest transactions first).

Print the SQL queries for a-c and the result for c.

SELECT

TO\_CHAR(LAST\_MODIFIED\_DATE, 'YYYY-MM-DD HH24-MI-SS'),

PATRON\_ID,

LAST\_NAME,

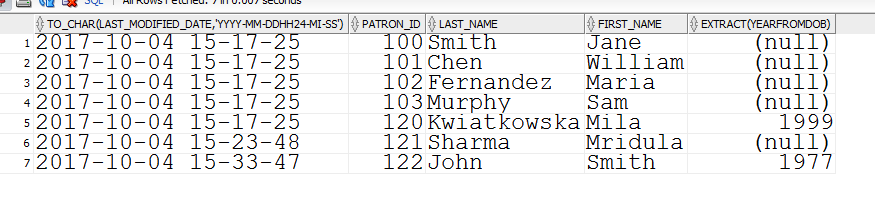
FIRST\_NAME,

EXTRACT(YEAR FROM DOB)

FROM L1\_PATRONS

WHERE LAST\_MODIFIED\_DATE > SYSDATE -10

ORDER BY LAST\_MODIFIED\_DATE; -- note: 10 is in days



**Part 2** Data Modeling for the TRU Sleep Clinic (2 %)

Use the following information/knowledge to prepare data requirements for the TRU Sleep Clinic.

TRU sleep clinic runs a special study on blood pressure (BP) and sleep disorders (specifically obstructive sleep apnea). The following data has to be kept for each patient: patient ID, gender, DOB, date/time of assessment, weight, height, BMI (calculated from weight and height), blood pressure: systolic and diastolic BP, antihypertensive medication (yes/no). The data for the weight, height, BP, and AH medication are recorded once a month. The weight and height should be stored using the International System of Units (SI standard/ metric).

1. Create a **Logical Data Model** (use Relational Model) for the above study: list the relations with attributes, PKs, and FKs. For each attribute provide a short description (meaning), domain, standard for units/representations. Patient ID is a unique integer number assigned by the Clinic Coordinator. Gender should allow for undeclared values (represented by ‘U’) and unknown (missing values) as NULL.

List the relations with indicated PKs (underlined) and FKs (in italics).

PATIENT(PATIENT\_ID, GENDER, DOB)

ASSESSMENT(ASSESSMENT\_ID, ASSESSMENT\_DATE\_TIME, WEIGHT\_KG, HEIGHT\_CM, BMI, SYSTOLIC\_BP, DIASTOLIC\_BP, AH\_MEDICATION, *PATIENT\_ID*)

PATIENT\_ID number assigned by the clinic coordinator NOT NULL primary key

GENDER char(1) Domain = {M, F, U} NULL allowed

DOB date Valid date (time will be set to 00)

…

1. Using your Logical Data Model, create a **Physical Data Model**. Prepare a **DDL script to create the table(s) with the appropriate constraints (entity, referential integrity, and domain)**. The BMI column should be defined as a virtual column (value generated automatically based on other columns). Calculate the Body Mass Index (BMI) of the patients at the time of the study. Use only as many significant digits as is justified. **BMI = weight(kg)/height(m)2**

**Print the DDL script.**

**CREATE TABLE patients**

**(patient\_id NUMBER (10) PRIMARY KEY,**

**gender CHAR(1)**

**CHECK (gender IS NULL OR gender IN ('M','F','U')),**

**DOB DATE)**

**CREATE TABLE assessments**

**(assessment\_id NUMBER (10) PRIMARY KEY,**

**assessment\_date\_time DATE NOT NULL,**

**weight\_kg NUMBER (3) NOT NULL,**

**height\_cm NUMBER (3) NOT NULL CHECK (height\_cm > 10),**

**systolic\_bp NUMBER (3) NOT NULL,**

**diastolic\_bp NUMBER (3) NOT NULL,**

**AH\_medication CHAR(3) CHECK (AH\_medication IN ('YES', NO')),**

**bmi AS (weight\_kg/((height\_cm/100) \* (height\_cm/100))),**

**patient\_id number (10) references patients(patient\_id));**

To supply a value for DOB column you can either use the TO\_DATE() function or standard ANSI/ISO date literal e.g., **DATE '1999-09-29' (note the date format must be YYYY-MM-DD)**

1. Using your table(s) created in 2 perform the following tasks:
2. Write SQL DML statements to add **3 patients**. For each patient add data for **3-4 clinical assessments.**

**INSERT INTO patients VALUES (1,'F', DATE '1999-09-29');**

**INSERT INTO patients VALUES (2,'M', DATE '1965-08-29');**

**INSERT INTO patients VALUES (3,null,DATE '1979-06-01');**

**--just an example**

**INSERT INTO ASSESSMENTS**

**(**

**ASSESSMENT\_ID,**

**ASSESSMENT\_DATE\_TIME,**

**WEIGHT\_KG,**

**HEIGHT\_CM,**

**SYSTOLIC\_BP,**

**DIASTOLIC\_BP,**

**AH\_MEDICATION,**

**PATIENT\_ID**

**)**

**VALUES (1212,SYSDATE, 68, 175, 120, 80,'YES', 3 );**

1. Write an SQL query to list the patients and their age at the time of each assessment. List the patient ID, age, and assessment date.

**PLEASE NOTE: MULTIPLE ANSWERS POSSIBLE**

-- rounding or truncating to the nearest integer value

-- the difference between two dates is in days

SELECT PATIENT\_ID

,ROUND(((ASSESSMENT\_DATE\_TIME - DOB)/365)) "Age"

,ASSESSMENT\_DATE\_TIME "Assessment Date"

FROM ASSESSMENTS NATURAL JOIN PATIENTS;

select PATIENTS.PATIENT\_ID

,ROUND(((ASSESSMENT\_DATE\_TIME - DOB)/365)) "Age"

,ASSESSMENT\_DATE\_TIME "Assessment Date"

from ASSESSMENTS join PATIENTS

on ASSESSMENTS.PATIENT\_ID = PATIENTS.PATIENT\_ID;

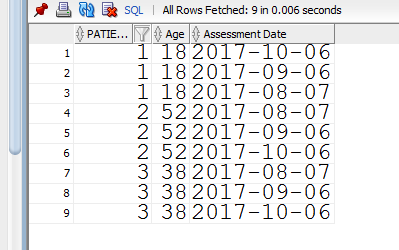
select P.PATIENT\_ID

,ROUND(((ASSESSMENT\_DATE\_TIME - DOB)/365)) "Age"

,ASSESSMENT\_DATE\_TIME "Assessment Date"

FROM ASSESSMENTS A, PATIENTS P

where A.PATIENT\_ID = P.PATIENT\_ID;



1. Write an SQL query to list all patients’ BMI characteristics: min, max, median, mean, and standard deviation.

-- Note rounding function

SELECT ROUND( MIN(BMI),1) "Min" ,

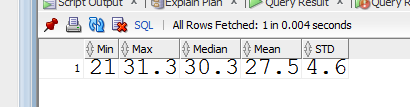
ROUND(MAX(BMI),1) "Max" ,

ROUND(MEDIAN(BMI),1) "Median" ,

ROUND (AVG(BMI), 1) "Mean" ,

ROUND (STDDEV(BMI),1) "STD"

FROM ASSESSMENTS;



**Print the SQL statements for a, SQL queries and results for b and c.**

**Part 3** Research (1 %)

**Using online documentation for MySQL 5.7 and Oracle 12c, answer the following questions:**

1. Compare and contrast the methods for creating an **automatic unique identifiers** in each DBMS. Put your answer in a short point/paragraph/table format. List references (at least one source in APA style). Check the library for APA style guides (ask the librarians), for example <http://libguides.tru.ca/APA>
2. Compare and contrast the **storage of date, time (precision), and time zones.** Specify the data types for storing date, time, and time zones in each DBMS. How dates are stored internally in each DBMS? What is the maximum range for dates? Prepare your answer in a table format. List references (at least one source in APA style).

**Submission of the Assignment**

Please create a pdf document with your answers. Make sure that the queries are printed using a **monospaced font (e.g., Courier font)** and are **properly formatted** (see textbook). Create a pdf document and upload the document to the BLearn. Indicate the Part and Problem number for your answers. Your document should have a title page with your name, student number, course title, and date. Upload pdf file.