**COMP 3610 Final Review**

**During the final exam the students will be allowed to use the following materials:**

* **One page of PL/SQL and SQL syntax will be included in the exam’s material (posted on BLearn for preview)**

**Please review the material from both books and class notes.** There are two copies of the Connolly&Begg book and two copies of Oracle 11g SQL book on the reserve in the Main Library.

The students should be familiar with the core SQL statements. The PL/SQL questions will cover a creation of relatively short function, trigger, and stored procedure. The conceptual, logical, and physical modeling is very important. Please review ERD, relational schema, normalization. The support for OO in SQL (UDT) will be tested at introductory level (create UDT and use for the creation of a table).

**Overview of the material to be covered by the final exam**

1. Introduction to database concepts Connolly&Begg

* **Chapter 1.** Intro to Databases
* **Chapter 2.** Database Environment
* **Chapter 3.** Database Architectures and the Web

2. Relational model Connolly&Begg **Chapter 4**

* Explain the main terms used in the **relational data model**: relation, attribute, domain, tuple, degree, cardinality.
* Identify the three types of the keys (candidate, primary, alternate) and composite keys.
* Identify the relational integrity constraints (domain, entity, referential) for a small database.
* Design and analyse a **relational schema** for a small database.
* Explain the purpose of nulls in the context of a data model.
* Describe the purpose of a **view** in a database.
* Contrast the roles of a base table and a view.
* Determine the **degree** and the **cardinality** of a relation.
* NOTE: remember about the **textual representation** of relational schema (p.111 in DS book)

Relation1(attribute1, attribute2, …)

Relation2(attribute1*, attribute3*,…)

3. Relational Algebra Connolly&Begg **Chapter 5 (only Relational Algebra 5.1)**

* Form queries in the relational algebra and corresponding queries in SQL.
* Perform unary relational algebra operations on a given relation (instance of a relation). List the results.
* Perform set operations on two relations (instances of relations). List the results.
* Perform Cartesian product operation on two sets.
* Determine the cardinality of the results.
* Determine if two relations are union-compatible.
* Perform join operation on given relations (inner join, outer join, semi join).
* Write an SQL query corresponding to a given relational algebra expression.

1. SQL Oracle 11g book and Connolly&Begg Chapter 6 and 7.

***Please note that SQL used in Oracle is a “dialect” of SQL. Chapters 6 and 7 cover ISO Standards for SQL. The exam requires practical skills in writing SQL Oracle 12c statements (Oracle 11g).***

**SQL: Basic SELECT statement** SQL book Chapter 1 & 2

* Explain the role of ISO SQL standards and the implementation of standards in different DBMSs.
* Explain why SQL is called a declarative (non-procedural) language.
* Understand **BNF notation** used to define SQL syntax.
* Explain the role of give examples of reserved words in SQL.
* Write SQL queries using current version of Oracle DBMS.
* Write simple subqueries and **correlated subqueries** (e.g., EXISTS)
* Write SELECT statements using arithmetic operations and concatenation for strings.
* Write SELECT statement using basic regular expressions (the syntax will be provided)
* Correct syntactical errors.
* Correct logical errors (use correctly logical operators AND, OR, and IS NULL).

**SQL: Data Manipulation SQL book chapter 5**

* Add a row of data to an existing tableusing an **INSERT** …VALUES statement.
* Add row(s) of data using existing table **INSERT …SELECT**
* Change data values using **UPDATE.**
* Delete data using **DELETE.**
* Use transaction control statements **COMMIT and ROLLBACK.**

**SELECT** statement **SQL book Chapter 8**

* Use **WHERE** clause to retrieve subset of rows (relational operation of selection).
* Sort the results using **ORDER BY** clause.
* Use queries with operators IN, LIKE, BETWEEN.

**Joining Tables and Combining Results SQL book Chapter 9** Connolly&Begg

Chapter 6.3.7 Multiple-table queries

* Write multi-table queries using inner joins using traditional syntax and SQL 1992 standards.
* Write queries using **OUTER JOIN**s**.**
* Explain the use of set operators in SQL.
* Map the algebraic set operations to SQL queries with UNION, INTERSECT, and MINUS.
* Write queries using Oracle 11g set operators**: UNION, INTERSECT, MINUS.**

**Scalar Functions SQL book Chapter 10**

* Write queries to perform **string** operations using built-in functions**: LOWER,UPPER,INITCAP,SUBSTR,**

**LENGTH, LPAD, RPAD, RTRIM, LTRIM, CONCAT ||**

* Write queries to perform **temporal** operations using date and time on the server: SYSDATE, SYSTIMESTAMP and client session: CURRENT\_DATE, CURRENT\_TIMESTAMP.
* **Format date and time** using **ISO standard** and local formatting: TO\_CHAR, TO\_DATE.
* Calculate **dates** using built-in functions: ROUND, TRUNC, TO\_DATE, MONTHS\_BETWEEN, LAST\_DAY, NEXT\_DAY, ADD\_MONTHS.
* Difference between CURRENT\_DATE and SYSDATE
* Regular expressions: REGEXP\_LIKE REGEXP\_SUBSTR use and understanding of short expressions.
* Write queries to perform numeric calculation using numeric functions: ROUND,TRUNC, TO\_NUMBER.
* Write queries to perform calculation on columns with NULLs using NULL replacement function: NVL, NVL2.

**Aggregate (Group) Functions SQL book Chapter 11** (skip GROUPING SETS, CUBE, ROLLUP))

* Write queries to group rows and to calculate aggregate functions: **AVG, COUNT,MAX, MIN, SUM**.
* **GROUP BY …. HAVING**
* Difference between **COUNT(\*) COUNT(DISTINCT** *colname***) COUNT (colname)**
* Understand calculations with Nulls in aggregate functions.

**Subqueries** SQL book Chapter 12 (skip MERGE statement)

* Write nested queries using **un-correlated subqueries**.
* Write nested queries using **correlated subqueries** and **EXISTS** and **NOT EXISTS** keywords.
* Write DML statements using subqueries
* Use subqueries with IN and NOT IN.

**Data Definition Language DDL: TABLE SQL book Chapters 3,4,6,13**

* Determine the appropriate data type for storing data in a DBMS.
* Identify Oracle 11g data types for storing strings, numbers, date and time, time intervals, long texts, character objects CLOB, binary objects BLOB.
* Write DDL statements to create tables and constraints in your Oracle Schema.
* Add constraints to an existing table: PRIMARY KEY, FOREIGN KEY, NOT NULL, CHECK.
* Write DDL statement to create a table with a virtual column.
* Use CREATE TABLE statement and create a virtual column (calculated value).
* Write DDL statements to modify or drop an existing table: **ALTER** , **DROP**, **TRUNCATE.**
* Skip **FLASHBACK** and **PURGE** to retrieve or remove a table from the recycle bin.
* Write DDL statement to create a table from existing table(s) using **CREATE TABLE .. AS.**
* Write DDL statement to rename a table.
* Write SQL query to search data dictionary **USER\_TABLES**, **USER\_OBJECTS, USER\_CONSTRAINTS** to list the tables and constraints you have created in your schema.

**DDL: SEQUENCE** Explain the role of a sequence to generate sequential numbers for primary keys in Oracle DBMS.

* Explain the role of CYCLE/NOCYCLE and CACHE/NOCACHE options in the sequence.
* Compare and contrast the use of sequence in Oracle DBMS with the use of an AUTONUM data type in MS Access.
* Write DDL statements to create sequences.
* Use sequence to add a new value in an INSERT statement (NEXTVAL).

**DDL: INDEX**

* Explain the purpose of the secondary indexes.
* Write DDL statements to create indexes.
* Explain the difference between B-trees and Binary trees.

**DDL: VIEW**

* Describe the advantages and disadvantages of views.
* Explain how the defining query is used in the CREATE VIEW statement.
* Explain the importance of the view constraints: WITH CHECK OPTION and WITH READ ONLY.
* Write DDL statements to create views.
* Perform DML operations on simple views.
* Explain the difference between simple views and complex views in the context of view updatability.

**5. PL/SQL Language** Connolly&Begg Chapter 8 Advanced SQL (skip recursion)

* Write short PL/SQL programs to create a simple function
* Write a function to calculate date, # of days, age (based on DOB), …
* Design, write code, and test a simple stored procedure.
* Write a stored procedure to access multiple rows (CURSOR) and display data.
* Use simple cursor with FOR loop.
* Design, write code for a simple trigger.
* Use trigger and sequence for adding numbers (numbers and strings) to a primary key.
* Use triggers to stop update/insert/delete based on a business rule (RAISE\_APPLICATION\_ERROR)
* Define integrity constraints in a procedural way using database triggers.
* Advantages and disadvantages of triggers.

**6. Database Analysis and Design** Connolly&Begg Chapter 10 and 11

* Identify the main stages and activities in the database development lifecycle

**7. Conceptual model: ERD and EERD** Connolly&Begg Chapter 12 and 13

* Use **UML notation** and **crow’s feet notation** ERD for a small database model.
* Interpret an existing ERD. Create an ERD for a small system. Draw an EERD for a small database system.
* Explain and demonstrate the concepts of attribute inheritance in superclass/subclass relationships.
* Give examples and represent using EERD participation and disjoint constraints.
* Give examples and represent using EERD an aggregation and a composition.

**8. Normalization** Connolly&Begg Chapter 14

* Identify candidate keys, primary keys, and alternate keys.
* Identify multi-valued and composite attributes.
* Identify **functional dependencies**.
* Normalize a 1NF relation into a set of 3NF relations (decompose relation).
* **Identify relations not in 2NF, 3NF and general normal forms.**

**9. Conceptual/logical/Physical database design** Connolly&Begg Appendix D

* Discuss the benefits of indexes. Discuss the “cost” of indexing all attributes for a highly volatile table.
* Discuss (give examples) of **atomicity** in context of the ACID properties of a transaction?
* Explain the AUTOCOMMIT setup.

**10. Introduction to OODBMS and ORDBMS** Connolly&Begg Chapter 9 (Please note 9.6 relates to OO extensions in Oracle) Notes on UDT

* Definitions for BLOB, CLOB, NCLOB.
* Describe the difference between RDBMS, OODBMS, and ORDBMS. Give examples for each.
* List two benefits of using Objects
* **User Defined Types** (UDT) in Oracle CREATE TYPE, CREATE TYPE .. AS OBJECT
* Oracle support for semi-structured data: Storing XML documents in Oracle (XMLTYPE)
* Oracle support for semi-structured data: Storing JSON document in Oracle.

**EXERCISES**

**Please review midterm I and II and the review materials. Please review Assignment 1-3 and LABS answers (posted on the BLearn)**

**Problem 1:** The following tables form part of a database held in a relational DBMS. This is a database for one hotel only:

**Room (roomNo, type, price)**

CREATE TABLE FE\_ROOM(

roomNo NUMBER(3) PRIMARY KEY,

r\_type VARCHAR2(15) NOT NULL,

price NUMBER(6,2) NOT NULL

);

**Booking (bookingNo, guestNo, dateFrom, dateTo, roomNo)**

CREATE TABLE FE\_BOOKING(

bookingNo NUMBER(6) PRIMARY KEY,

guestNo NUMBER(6) NOT NULL,

roomNo NUMBER(3) NOT NULL,

dateFrom DATE NOT NULL,

dateTo DATE NOT NULL,

CONSTRAINT guest\_booking\_fk\_fe FOREIGN KEY(guestNo) REFERENCES fe\_guest(guestNo),

CONSTRAINT room\_booking\_fk\_fe FOREIGN KEY(roomNo) REFERENCES fe\_room(roomNo)

);

**Guest (guestNo, guestName, guestAddress)**

CREATE TABLE FE\_GUEST(

guestNo NUMBER(6) PRIMARY KEY,

guestName VARCHAR2(35) NOT NULL,

guestAddress VARCHAR2(60) NOT NULL

);

**ROOM INDEX ON TYPE AND COST**

CREATE INDEX fe\_room\_index ON FE\_ROOM(fe\_roomtype,fe\_room\_cost) COMPUTE STATISTICS;

**GUEST SEQUENCE**

CREATE SEQUENCE guest\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 0

MAXVALUE 100;

**GUEST TRIGGER**

CREATE OR REPLACE TRIGGER FE\_AUTO\_INCREASE

BEFORE INSERT ON FE\_GUEST

FOR EACH ROW

BEGIN

:new.guestno := guest\_sequence.nextVal;

END;

**GUEST INSERT**

INSERT INTO FE\_GUEST

(GUESTNAME,GUESTADDRESS)

VALUES('GUEST TEN','ADDRESS TEN');

1. Create tables, sequences, indexes, triggers for sequences.
2. Create a PL/SQL procedure to list the guests (guestNo, guestName) staying tonight in our hotel.
3. Write a query to list all rooms available for one night tonight (SYSDATE).
4. Create a view ECONOMY\_ROOMS which will show only the rooms with price below $100.00.
5. Create a PL/SQL procedure TOP\_GUESTS (MIN\_STAYS NUMBER, YEAR NUMBER) with two parameters minimum number of stays and a year (as a number). This procedure will list all guests (guestNo, guestName) who has stayed (have bookings) for at least the minimum number of stays in a given year. “A booking in 2016” means that the booking starts and ends in 2016.
6. Add a new guest (add yourself) and a booking for two days starting today (SYSDATE) in roomNO 133.
7. Reverse engineer the physical model and draw an ERD
8. **Relational algebra**: Write a relational algebra expression to list the guests who have bookings for December 24, 2016. This means they will be staying in our hotel at that day.
9. The hotel manager plans to create a special reward plan for the guests who have more than 5 bookings in 2016. List the guests (guestNo, guestName) and their total number of bookings in 2016. Include only the guests who have more than 5 bookings. “A booking in 2016” means that the booking starts and ends in 2016.
10. List the guests (guestNo, guestName) and their **total number of days booked** in 2016. Include only the bookings starting and ending in 2016.
11. Calculate the **average length of stay** (in days) for 2016. Include only the bookings starting and ending in 2016.
12. Create a **function** to calculate the utilization of rooms total# of booked days/ (total# of rooms \* 365) \* 100% for a given year (year is the parameter value).
13. Write a DDL statement to add a new column BOOKING\_DATE to Booking. This column will be used to store the date/time of the booking.
14. The booking clerk wants to study the patterns of bookings. She is planning to introduce a special booking discount for the bookings done on Saturdays and Sundays. Using the redesigned table (from the previous question) write an SQL **SELECT statement** to calculate the total number of bookings done on Saturdays and Sundays in 2016.
15. The booking clerk wants to know how the bookings are done. In general, the bookings can be done online via the Web system or via the phone. Write a DDL statement to add a new column BOOKING\_TYPE the valid values are “web” and “phone’. Since this column will be added to the existing table (with data) make sure that NULLs are allowed.
16. Using the new columns list the total number of bookings done on weekdays (Mon through Fri) between 9:00 and 17:00 via phone. Exclude from calculation NULLs (we don’t know the days/time/type).
17. Write a DML statement to add a booking for yourself (use a hard-coded GuestNo) for January 2, 2017 for 2 nights. You are making the booking today (SYSDATE) using Web. All rooms are available, so book yourself in any room (hard-coded value for roomNo).

**Problem 2:** The following tables form part of a database held in a relational DBMS:

Employee (EmpNo, EmpName, Hire\_Date, Gender, Salary, DepNo, Supervisor\_Empno)

Department (DepNo, DepName)

1. List all employees working for the “Research” department.
2. List the employees working for the “Research” department longer than 3 years. Assume that the employees work only for one department throughout their employment.
3. Assume that some Employees are not assigned to a department. How this fact is represented in the database? Be very specific. Write a query to list all Employees (EmpNo, EmpName) who are not assigned to a department.
4. Find the employee(s) (EmpName, Hire\_Date (using ISO standard), Salary (using $999,999 formatting) with the highest salary.
5. List the employee(s) (EmpNo) whose salaries are higher than the average salary in the same department.
6. Write one SELECT statement to list the average salary for females (Gender = ‘F’) and the average salary for males (Gender = ‘M’). Hint, the solutions are many, but a relatively easy is using UNION. The other solution is the use of the subquery in the SELECT clause, for example the following query is counting names starting from A and names starting from J:

SELECT

(SELECT count(EmpNo) FROM employee WHERE upper(EmpName) LIKE 'A%') "A",

(SELECT count(EmpNo) FROM employee WHERE upper(EmpName) LIKE 'J%') "J"

FROM DUAL;

1. Write a DML statement to add a new department (DepNo is added automatically by an existing trigger). The Department name is “Data Analysis”.
2. Create a **function** YEARS\_HIRED to calculate the number of the years of employment given the Hire\_date and today’s date (assume SYSDATE).
3. The EmpName column holds the first and the last name of the employee. This is not the best design, but it was done that way. However, the data is consistent the first name is always first, followed by one space, and last name. Using the regular expression find the last names of all employees.
4. Write a DML statement to add yourself as an employee. Employee number is automatically generated by an existing trigger. You will be working in the “Data Analysis” department (use a subquery to add the appropriate Department Number). Give yourself a nice salary and use SYSDATE for Hire\_Date.
5. List (Department Name, Average Salary) the average salary in each department. Sort by the Department Names.
6. Create a trigger to raise an error when salary is updated on Sunday.
7. Create an AUDIT table for EMPLOYEE table and a trigger to write all updates involving Salary. The audit data should include: date and time of change, user id, employee no, and salary before and after change.
8. Increase the salary by 10% for all employees working longer than 10 years in “Research” Department.
9. List all employees (EmpNo, EmpName) and their immediate supervisor’s name. Note: there is one employee who is the top CEO and has no supervisor (include him/her in the list).
10. List the supervisors (the EmpNo of the supervisor and the EmpName of the supervisor) and the number of the directly supervised employees (this number will be at least 1).
11. Re-write the query from 14 to include all employees (the supervisors and the ones who are not supervisors). This should be a small change. Note: Use COUNT
12. Write a DDL to add a constraint for an existing table on the column Salary. A salary must be between 100 and 100,000.
13. The HR just realized that they need to keep the data of the employees’ dependents. Write a DDL statement to add a new table DEPENDENT to keep data on the employee’s dependents. The following data are required: Dependent Name, DOB, and Gender. Create the new table with the primary key constraint and the foreign key constraint.
14. Using the newly created table DEPENDENT, list all employees (the column titles: Employee Number, Employee Name, Number of Dependents) who have more than two dependents.
15. The “Research” Department is planning a Christmas party for all their employees and their dependents who are 12 years or younger. Write a select statement to calculate the total number of children who will be invited to this party.
16. Let’s assume that the employees may work for multiple departments. How would you change the database schema to handle this requirement? (many to many relationship question) Write the DDL statements to create database objects required to store the appropriate data.
17. Let’s assume that the user of the system asked you to maintain the history of employment for each department. The user wants to know who and when was working for a specific department. How would you have to change the schema? (temporal data question).

**Problem 3. Relational Algebra Exercise**

Given two relations (tables) **R**  and **S:**

**R**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **D** |
| a1 | b1 | c2 | d1 |
| a3 | b1 | c1 | d2 |
| a2 | b2 | c4 | d5 |

**S**

|  |  |
| --- | --- |
| **E** | **A** |
| e1 | a3 |
| e3 | a2 |

Answer the following questions:

1. What is the result of a Selection **σ**  over **R** :**σ** B <> b2 (**R**) =

**B <> b**2 means that *the value in column B is not equal to b2*

1. What is the result of a Selection over R: **σ** B <> b2 and B<> b1 (**R**) =
2. What is the result of a Projection over **R**: **π** B (**R**) =
3. Is Union **R ∪ S** a valid operation? Explain why.
4. What is the **degree** (number of columns) of the result **R x S** (**x** – Cartesian Product)?\_\_What is the **cardinality** of the result?\_\_\_\_
5. Calculate **natural join** between R and S. What is the degree of the R ⊳⊲ S?
6. Calculate Left Outer Join between R and S.
7. What is the result of **R ∩ S** ?

**Problem 4:**

The following table represents data for the appointments in a driving school We-teach-U-how-2-drive. All appointment are for 1 hour.

1. The data in the table is susceptible to update anomalies. Provide examples of how insertion, deletion, and modification anomalies could occur on this table.
2. Identify the functional dependencies represented by the data shown in the table. State any assumptions you make about the data.
3. Draw a **conceptual model** for the database to represent the above data.
4. Map the conceptual model (ERD) into a logical model (relational). List the relations with primary and foreign keys.
5. Normalize (up to 3NF) the relations if required (most likely the relations will be in 3NF).
6. Create a physical model for the driving school database. Write CREATE statements to create the APPOINTMENTS table (assume that the other tables have been already created).

**Problem 5:**

1. What is the role of the SEQUENCE object in the Oracle database?
2. What is the difference between DATE and TIMESTAMP data types in Oracle 12c?
3. Why dates should be stored using special data types for dates? Why dates should not be stored as a string of characters?
4. What is the difference between a candidate key and a primary key?
5. In 1970, E. Codd introduced a new data model for databases. What was the name of that model?
6. Why Oracle 11g can be called an ORDBMS, but not OODBMS?
7. What is the difference between these two data types: CHAR and VARCHAR2?
8. The DDL language allows for creation of the database objects. The DML language allows for the following four operations: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
9. SQL is a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(declarative/procedural) language, because SQL states what the data is needed rather than how it is to be retrieved.
10. The relational data model is based on the concept of mathematical relations. In the relational model, the **relationships between entities are represented** by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
11. We discussed in class the difference between the conceptual model and logical model. The textbook makes also a distinction between these two models. What is the underlying difference between a conceptual data model and a logical data model?
12. What is the function of a system catalogue in a DBMS? In Oracle DBMS the system catalogue is often called a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
13. In the relational mode, each attribute must have a domain, and more than one attribute may use the same domain\_\_\_\_\_\_(True/False)
14. In relational model, when a key consists of more than one attribute, it is called a \_\_\_\_\_\_\_\_\_\_\_key.
15. Why Nulls are needed in the databases?
16. Give an example of a CASE software.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
17. What is the difference between CLOB and BLOB?
18. Write a command to change your user password for your Oracle account to “XYZ”
19. Explain the result of the following statement

GRANT SELECT ON patients TO PUBLIC;

1. What are the results of the DESCRIBE patients command after the following DDL statements:

DROP TABLE patients;

TRUNCATE TABLE patients;

1. What is the difference between the following two DDL statements:

DROP TABLE patients; and DROP TABLE patients PURGE;

1. The four basic properties of transactions are called ACID (Atomicity, Consistency, Isolation, Durability). Which of this properties is defined below

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_means that a series of database operations either all occur, or nothing occurs.*

**Problem 6:**

1. Using UML standards, draw an **EERD** (include primary keys, attributes, relationships, participation constraints, disjoint constraints, and cardinalities) for the following university application.

ABC university has a large number of employees. Each employee has an employeeID, name, address, and SIN. There are exactly three types of employees: faculty employees, librarians, and research assistants. Faculty employees have monthly salaries, and teach courses. Each faculty employee teaches one or up to four courses in a given semester (Fall, Winter, Summer) and year. Each course has course ID (e.g., 2016023610001), code (e.g., COMP3610) and a course name (e.g., Database systems). Each course may be offered multiple times. Each offering of the course is taught by one faculty. Each librarian works on at least (possibly many) special tasks (taskID, description). Special tasks have none or one or many librarians. Each research assistant (studentID, program of study) works on at least one project (possibly many). Project might be assigned to many or none research assistants. Projects have project ID, description, start and end date, and budget (amount in dollars).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **appNo** | **date/time** | **instructorID** | **iFName** | **iLName** | **clientIDD** | **cFName** | **cLName** | **cAddress** |
| 1001 | 2016-10-12 09:00 | I456 | Jane | Watt | C034 | Anne | Way | 111 Storrie Road, Paisley |
| 1102 | 2016-10-12 10:00 | I456 | Jane | Watt | C034 | Anne | Way | 111 Storrie Road, Paisley |
| 1203 | 2016-10-12 19:00 | I344 | Tom | Jones | C034 | Anne | Way | 111 Storrie Road, Paisley |
| 1334 | 2016-11-12 13:00 | I666 | Karen | Black | C089 | Mark | Fields | 120 Lady Lane, Paisley |
| 1455 | 2016-11-12 14:00 | I957 | Steven | Smith | C019 | John | Brown | 13 Renfrew Road, Paisley |
| 1676 | 2016-11-12 14:00 | I344 | Tom | Jones | C039 | Karen | Worth | 34 High Street, Paisley |

1. Create a logical model (relational) using the above EERD.

**Problem 7:**

Use the following relational schema. Please note that our Hotel chain is growing and now we have hotels in many cities:

HOTEL (Hotel\_ID, H\_Name, H\_City)

ROOM (HotelID, RoomNo, Room\_Type, Room\_Cost)

ROOM\_RESERVATION (ResNo, DATE\_FROM, DATE\_TO, C\_ID, HotelID, RoomNo)

CLIENT (C\_ID, C\_First\_Name, C\_Last\_Name, C\_Address, C\_City, C\_DOB)

1. Write a CREATE statement for the **ROOM** table. Room type is stored as a single character, for example, ‘D’ for double, ‘S’ for single, etc. Room Cost is in $ per day (maximum $5,000.00). Include the specification for the **primary key** and **referential integrity** constraints. Include the **constraint for the maximum** cost. The HOTEL table has been already created, and the HotelID domain is NUMBER(10).
2. Write a select statement to list all clients (C\_ID only) who celebrate their birthdays today (using the date on the server as today’s date) and whose last names have at least 12 characters. Hint: The LENGTH function returns the number of characters in a string.
3. The manager of the “Hotels for everybody” wants to analyse the demographics of the clients. He/she asked you to prepare the demographics report. The report will show city (from client’s address) and the minimum, maximum, and average age in years. Also, he/she has asked you to change the database to be able to add the client gender. How would you do it?

1. Write an SQL statement to add a new client to the CLIENT table. Primary key, C\_ID is automatically generated by an existing sequence C\_ID\_SQ. Client name is Jonathan O’Reilly, there is no data for address and city, and DOB is 1990-09-24. Hint: Remember about the ‘’ (**two single quotes**) for the second character in the O’Reilly name.
2. Write an SQL statement to increase the RoomCost by 15% for all rooms located in the hotels in Kamloops.
3. Write an SQL statement to list the hotels (id and name) in Kamloops and the total number of rooms for each hotel.
4. Write an SQL statement to list all reservations (dates, hotel id, hotel name) for the hotels in Vancouver for December 2016 (reservation may start or end in December 2016).

1. Write an SQL statement to delete all clients, who have never reserved a room.
2. Table ROOM\_RESERVATION has been created. The following SELECT statement lists all the constraints for this table. What is the result of the same query (below, when the table name is entered in a lower case?

select constraint\_type, constraint\_name

from user\_constraints

where table\_name = 'ROOM\_RESERVATION'

order by constraint\_type;

**Problem 8:**

Please use tables:

PATRONS, TRANSACTIONS, BOOKS (from SQLDeveloper’s tutorial)

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)

1. List the patrons (only patron\_id) and their total number of transactions (GROUP BY, COUNT).
2. Add the data to the Transactions table. Borrow a book (transaction type = 1) and return (transaction type = 2). List the transactions for this year (use SYSDATE) and group them by transaction type.
3. List (write query) the patrons (patron\_id, last\_name, first\_name) and their total number of transactions. This list will include only the patrons who have at least one transaction (INNER JOIN, GROUP BY… COUNT)
4. List (write query) all patrons (patron\_id, last\_name, first\_name) and their total number of transactions. Include also the patrons who have no transactions. (method1: OUTER JOIN method 2: UNION). Add data to test the query.
5. List the transaction (transaction\_id, book\_id, title) for the books with high rating (rating = 10) for the last 2 years (use SYSDATE).
6. Add two books by Joan Casteel “Oracle 11g SQL” and “Oracle 12c”.
   1. Write a query to list each author (first and last name) and their total number of books.
   2. Write a query to list only the authors (first and last name) who have more than one book.
7. List the books (book\_id, title) without transactions.

**Problem 9:**

1. Assume that the table VEHICLE has the following data:

|  |  |  |
| --- | --- | --- |
| V\_ID | MAKE | COST\_PER\_DAY |
| 9090 | BMW | 100.00 |
| 9111 | FIAT | 100.00 |
| 9191 | FORD | 51.00 |
| 9192 | FORD | 88.00 |
| 1001 | NISSAN | 40.00 |

How many rows will be deleted by the following SQL statements (after each statement there is a ROLLBACK executed):

DELETE FROM VEHICLE  
 WHERE UPPER(MAKE) IN (‘NISSAN’,‘ford’); \_\_\_\_\_\_?

DELETE FROM VEHICLE  
 WHERE MAKE = ‘NISSAN’ AND MAKE = ‘FORD’; \_\_\_\_\_\_?

DELETE FROM VEHICLE  
 WHERE MAKE = ‘NISSAN’ OR MAKE = ‘FORD’; \_\_\_\_\_\_?

DELETE FROM VEHICLE; \_\_\_\_\_?

1. The manger of KK-Car-Rentals asked you to “redesign” the VEHICLE table. Add the following columns: make (as a sting of characters), year (e.g., 2016), colour exterior, colour interior. Write a DDL statement to add the appropriate columns to the existing table VEHICLE to allow the car rental manager to enter specific vehicle’s data.
2. You have done outstanding work (see above), but your manager overheard somebody talking about UDTs in Oracle and demands that “his” database must use the modern standards. Create a UDT VEHICLE\_DETAILS\_TYPE to store the new data. Using your newly created UDT and updated table write an INSERT statement to demo your manager how he can use SQL to enter the following data: (V\_ID is 9999, MAKE is NISSAN, model is 370Z, year is 2017, colors are deep blue and black, cost per day $150.00.
3. Your manager wants you to add to the existing VEHICLE table a picture (jpg) of the car.

Write the DDL statement to add additional column to store pictures (jpg maximum size 2 MB).

**Problem 10:**

1. The relational model assumes both **horizontal** and **vertical homogeneity**. Using the PUBLISHER relation explain both terms.

PUBLISHER(PUBID, NAME, CONTACT, PHONE)

***Horizontal homogeneity*** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Vertical homogeneity \_\_\_***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Give an example of each DBMSs: RDBMS, ORDBMS and OODBMS.
2. Write an Oracle DDL statement to create an **object type** Address, which is composed of four attributes: street, city, province, country.
3. What is the meaning of the letter “g” in the name of **Oracle 11g**?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is the meaning of the letter “c” in the name of **Oracle 12c**?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Write a SELECT statement to display **the first day of the current month** (using SYSDATE) and time 9:00 in the morning. Use ISO standard for the date and time.

SELECT to\_char(TRUNC(SYSDATE,'MONTH'),'DD/MM/YYYY')||' 09:00' FROM DUAL;

1. Write a SELECT statement to display **the last day of the current month** (using SYSDATE) and time 9:00 in the morning. Use ISO standard for the date and time.

SELECT to\_char(ROUND(SYSDATE+10,'MONTH'),'DD/MM/YYYY HH:MI:SS') FROM DUAL;

**Problem 11:**

1. Is the following relation in the 3NF? Why yes or no? Is this relation in the GENERAL 3NF?

Vehicle(V\_ID, VIN, make, color, year) VIN is a Vehicle Identification Number

1. The following relation is not in 3NF. List the relations (3NF) after the decomposition.

Exam (ExamID, ExamDateTime, ExamCourseSection, InstructorId, InstructorName, InstructorPhone#)

1. Is the following relation in the 2NF? Explain why or why not.

Workload (InstructorID, CourseID, instructorName, courseTitle)

**Problem 12**

Use the following tables from a Veterinary database

**PETS\_COUNTS** (BREED\_NAME, BREED\_COUNT, LAST\_UPDATE\_DATETIME)

**PETS** (PET\_ID, PET\_NAME, BREED\_NAME, YEAR\_OF\_BIRTH)

The table **PETS** has been created and more than 100 pets have been added. However, the table PETS\_COUNTS hasn’t been created. Write DDL to create table PET\_COUNTS.

1. Write **one** DML to insert data into PET\_COUNTS using the PETS data and SYSDATE as LAST\_UPDATE\_DATETIME.

**Problem 14**

Tables: Products and Product\_types exist in the database, and were created by the following CREATE statements:

CREATE TABLE product\_types (

product\_type\_id INTEGER CONSTRAINT product\_types\_pk PRIMARY KEY,

type\_name VARCHAR2(10) NOT NULL

);

CREATE TABLE products (

product\_id INTEGER CONSTRAINT products\_pk PRIMARY KEY,

product\_type\_id INTEGER

CONSTRAINT products\_fk\_product\_types

REFERENCES product\_types(product\_type\_id),

Product\_name VARCHAR2(30) NOT NULL,

Product\_description VARCHAR2(50),

price NUMBER(5, 2) NOT NULL

);

1. Create a **function CALC\_AVG\_PRICE\_TYPE** to calculate an average product price for a given product type.
2. Write a query to list all products (product name, type name, price) and the average price for its type (use the function from 1).
3. Write a DDL to create two tables: ORDERS(order\_no, order\_date) and ORDER\_ITEMS(item\_id, product\_id, qty, order\_no). Specify the entity integrity and the referential integrity constraints. Specify the following business constraints: qty must be greater than 0 and not NULL; order\_date must be not NULL.
4. Write DDL to create an index to improve the performance of a typical join between Orders and Order\_items.
5. The store supervisor wants to see only the current orders (orders placed this moth). Your friend is creating a view to list current orders. However, you run the view and it lists the orders for one month but for all years. Fix the following code to list only the orders placed this month and this year.

CREATE VIEW current\_orders AS

select \* from orders

where extract(month from date\_ordered)= extract(month from sysdate);