Assignment 1 is to be completed at the end of Module 1. It is worth 4% of your final mark.

It is graded with a total of 20 marks, and each question is worth 5 marks. Submit an MS-Word document named COMP3611\_Assignment1 that contains your answers for this assignment.

Mengyang Zhang

T00696467

1. In your own words, describe the three generations of DBMSs. Research and briefly describe an example for each. (Do not re-state the examples mentioned in the textbook.)

The first generation DBMS is based on a hierarchical structure and the CODASYL model. The hierarchy is based on the concept that small components are grouped together into one large component and lasts until the final product is assembled. The CODASYL model was developed according to the proposal of DBTG, which specifies three different languages: DDL schema, DDL subschema and DML.

Example: HP's IMAGE Cullinet's Integrated Database Management System IDMS

The Second Generation of DBMSs is based on a relational model. The relational model is a table whose Columns are attributes and rows are records.

Example: Microsoft SQL Server, Oracle Database, MySQL and IBM DB2.

The Third Generation of DBMSs has two types:

object-oriented DBMS (OODBMS)

object-relational DBMS (ORDBMS).

However, the actual composition of these models are not clear.

Example: PostgreSQL and Oracle.

1. Research a job advertisement for a Database Administrator (DBA), Database Designer, or Database Application Developer. (Include the URL for the job advertisement in your answer.) Summarize the job description (1–2 sentences) and identify the typical work activities for the job (3–5 activities). Determine (if possible) the answer to the following questions:
   * What is the company?
   * What is the main purpose of the specific database application(s)?
   * Which DBMS (or DBMSs) are being used?
   * What are the required skills or certifications?
   * What is (approximately) the salary?
   * Are you personally interested in pursuing one of these career paths? Reflect on your career goals. How may this course help you in attaining them?

Senior Database Administrator

This job is mainly working on designing, developing and implementing Ministry database systems.

The working activities will be like deploying, administering and managing SaaS and PaaS Technology, doing engineering solutions to migrate databases from other platforms. Doing Spatial DBMS like ESRI SDE, Oracle Locator, etc. Administration of high availability solutions with PostgreSQL or ORACLE databases.

Salary Range: $72,724.97 to $83,014.85 plus 9.9% Temporary Market Adjustment.

Organization Env Assessment Office from BC Public Service

Requirement:

Degree, diploma, certification or equivalent in the computer science field.

A minimum of five years of experience with PostgreSQL and/or Oracle Enterprise Database Administration.

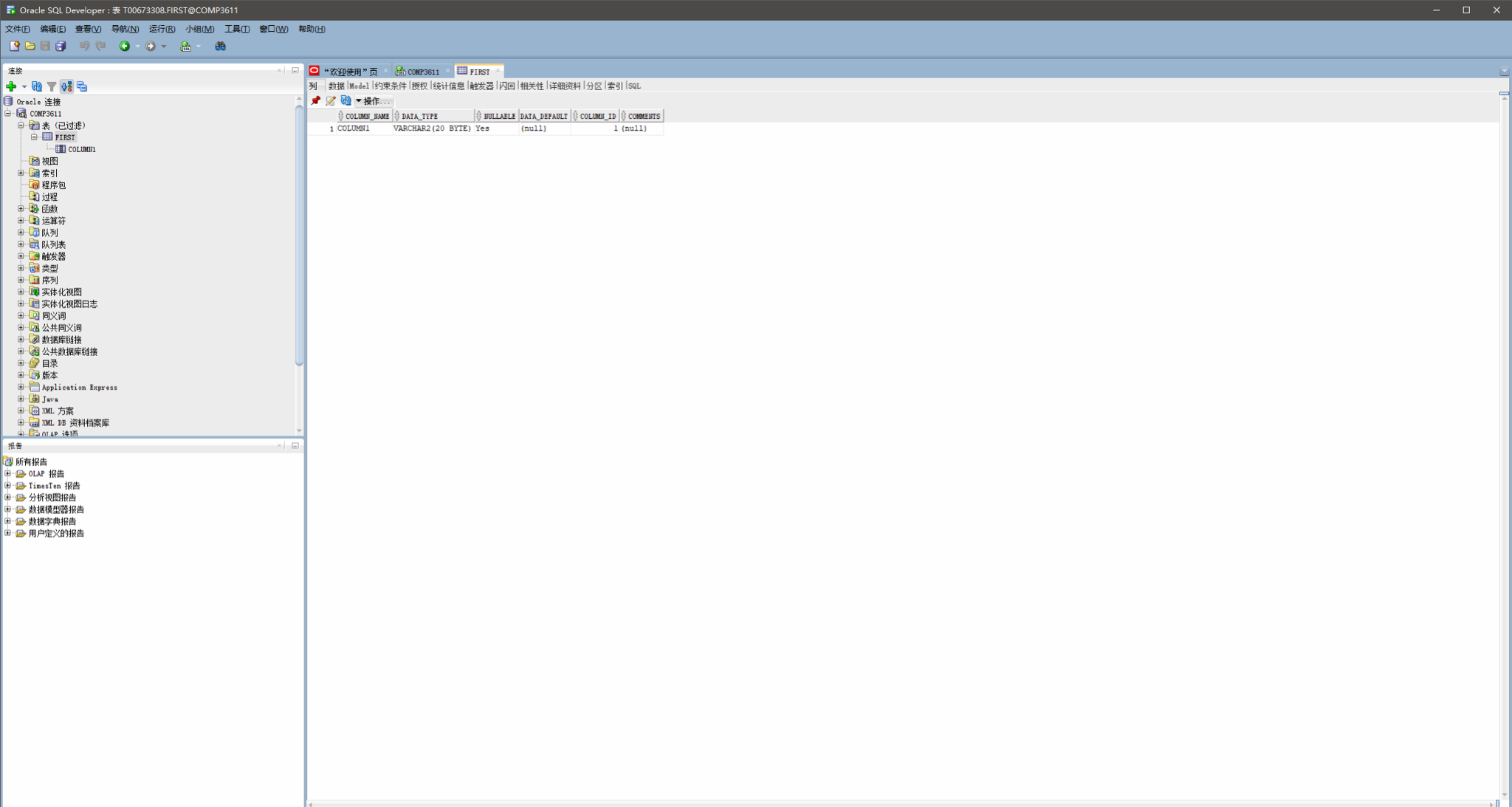
A minimum of two years of experience performing database backup and recovery procedures.

A minimum of two years of experience working with Structured Query Language.

A minimum of four years of experience working on implementing and/or supporting complex infrastructure projects.

<https://bcpublicservice.hua.hrsmart.com/hr/ats/Posting/view/77279>

1. Review the ten functions of a DBMS. Use SQLDeveloper to access an Oracle schema, and answer the following question:
   * How is function 2 (a user-accessible catalog) implemented?



it has COLUMN\_NAMEP : COLUMNI 1; DATA\_TYPE: VARCHAR2(20 BYTE); NULLABLE: Yes; DATA-DEFAULT : (null); COLUMN\_ID : 1; COMMENTS : (null)

1. Describe the main differences between OLTP and OLAP. Using the *DreamHome* case study in your textbook, give examples of several typical OLTP queries and OLAP queries.

online analytical processing (OLAP) and online transaction processing (OLTP). The main difference is that OLAP uses data to gain valuable insight, while the OLTP is purely operational. However, there are significant ways to use both systems to troubleshoot data issues.

The main distinction between the two systems is their names: analytical vs transactional. Each system is optimized for this type of treatment.

**OLAP:**

Using roll-up

SELECT name, address, type, SUM (Advance\_deposit) AS amount\_deposited FROM Lease

GROUP BY ROLLUP(name,address,type);

OLAP\_USER role

CREATE SESSION privilege

OLAP\_DBA role

CREATE SESSION privilege

OLAP\_XS\_ADMIN role

CREATE MATERIALIZED VIEW privilege

CREATE DIMENSION privilege

ADVISOR privilege

CREATE ANY MATERIALIZED VIEW privilege

CREATE ANY DIMENSION privilege

ADVISOR privilege

**OLTP:**

CREATE DATABASE InMemory

ON PRIMARY(NAME = InMemoryData,

FILENAME = 'd:\data\InMemoryData.mdf', size=200MB),o

-- Memory Optimized Data

FILEGROUP [InMem\_FG] CONTAINS MEMORY\_OPTIMIZED\_DATA(

NAME = [InMemory\_InMem\_dir],

FILENAME = 'd:\data\InMemory\_InMem\_dir')

LOG ON (name = [InMem\_demo\_log], Filename='d:\data\InMemory.ldf', size=100MB)

GO

USE InMemory

GO

-- Create a Simple Table

CREATE TABLE DummyTable (ID INT NOT NULL PRIMARY KEY,

Name VARCHAR(100) NOT NULL)

GO

-- Create a Memeory Optimized Table

CREATE TABLE DummyTable\_Mem (ID INT NOT NULL,

Name VARCHAR(100) NOT NULL

CONSTRAINT ID\_Clust\_DummyTable\_Mem PRIMARY KEY NONCLUSTERED HASH (ID) WITH (BUCKET\_COUNT=1000000))

WITH (MEMORY\_OPTIMIZED=ON)

GO

-- Simple table to insert 100,000 Rows

CREATE PROCEDURE Simple\_Insert\_test

AS

BEGIN

SET NOCOUNT ON

DECLARE @counter AS INT = 1

DECLARE @start DATETIME

SELECT @start = GETDATE()

WHILE (@counter <= 100000)

BEGIN

INSERT INTO DummyTable VALUES(@counter, 'SQLAuthority')

SET @counter = @counter + 1

END

SELECT DATEDIFF(SECOND, @start, GETDATE() ) [Simple\_Insert in sec] END

GO

-- Inserting same 100,000 rows using InMemory Table

CREATE PROCEDURE ImMemory\_Insert\_test

WITH NATIVE\_COMPILATION, SCHEMABINDING,EXECUTE AS OWNER

AS

BEGIN ATOMIC WITH (TRANSACTION ISOLATION LEVEL=SNAPSHOT, LANGUAGE='english')

DECLARE @counter AS INT = 1

DECLARE @start DATETIME

SELECT @start = GETDATE()

WHILE (@counter <= 100000)

BEGIN

INSERT INTO dbo.DummyTable\_Mem VALUES(@counter, 'SQLAuthority')

SET @counter = @counter + 1

END

SELECT DATEDIFF(SECOND, @start, GETDATE() ) [InMemory\_Insert in sec] END

GO

-- Running the test for Insert

EXEC Simple\_Insert\_test

GO

EXEC ImMemory\_Insert\_test

GO

-- Clean up

USE MASTER

GO

DROP DATABASE