



Xi'an Jiaotong-Liverpool University

西交利物浦大學

XJTLU Entrepreneur College (Taicang) Cover Sheet

Module code and Title	Database Development and Design (CPT201TC)	
School Title	School of AI and Advanced Computing	
Assignment Title	001: Assessment Task 1 (CW)	
Submission Deadline	17:00, 10th Dec (Friday)	
Final Word Count	NA	
If you agree to let the university use your work anonymously for teaching and learning purposes, please type "yes" here.		Yes

I certify that I have read and understood the University's Policy for dealing with Plagiarism, Collusion and the Fabrication of Data (available on Learning Mall Online). With reference to this policy I certify that:

- My work does not contain any instances of plagiarism and/or collusion.
- My work does not contain any fabricated data.

By uploading my assignment onto Learning Mall Online, I formally declare that all of the above information is true to the best of my knowledge and belief.

Scoring – For Tutor Use					
Student ID			1930080		
Stage of Marking	Marker Code	Learning Outcomes Achieved (F/P/M/D) (please modify as appropriate)			Final Score
		A	B	C	
1 st Marker – red pen					
Moderation – green pen	IM Initials	The original mark has been accepted by the moderator (please circle as appropriate):			Y / N
		Data entry and score calculation have been checked by another tutor (please circle):			Y
2 nd Marker if needed – green pen					
For Academic Office Use		Possible Academic Infringement (please tick as appropriate)			
Date Received	Days late	Late Penalty	<input type="checkbox"/> Category A <input type="checkbox"/> Category B <input type="checkbox"/> Category C <input type="checkbox"/> Category D		Total Academic Infringement Penalty (A,B, C, D, E, Please modify where necessary) _____



Xi'an Jiaotong-Liverpool University

西交利物浦大學

			<input type="checkbox"/> Category E	
--	--	--	-------------------------------------	--

Students

The assignment must be submitted in a single pdf document via Learning Mall Online to the correct drop box. Only electronic submission is accepted and no hard copy submission.

All students must download their file and check that it is viewable after submission. Documents may become corrupted during the uploading process (e.g. due to slow internet connections). However, students themselves are responsible for submitting a functional and correct file for assessments.



Xi'an Jiaotong-Liverpool University
西交利物浦大學

Database Development and Design (CPT201TC)

Assessment 001: Individual Coursework

Due: Dec 10th, 2021 @ 17:00

Weight: 50%

Maximum Marks: 50

Overview

This assignment aims to gain experience in understanding the internal functionality of different database management systems, including RDBMS, XML, Object-relational databases, and gain experience in designing data warehouses to perform OLAP operations. The course work will be assessed for the following learning outcomes:

- A. Identify and apply the principles underpinning transaction management within DBMS.
- B. Demonstrate an understanding of advanced SQL topics.
- C. Illustrate the issues related to Web technologies and DBMS and XML as a semi-structured data representation formalism.
- D. Identify the principles underlying object-relational models.
- E. State the main concepts in data warehousing and data mining.

Assessment Tasks

Your task is to answer every question by carefully reading the questions and guidelines for the system setup. Record your thoughts and assumptions, where necessary, while reporting your answers. There are the following five parts for this course work corresponding to five learning outcomes of this module:

1. Advanced SQL, Triggers, Indexing, and Query Optimization
2. Transaction Management
3. Querying XML Data
4. Object-Relational Database
5. Data Warehousing and OLAP

Marking Criteria

This coursework will be graded out of 50% of the total marks with a maximum of 50 marks. There are five (5) parts and ten (10) marks available for each part. Marks will be awarded as follows:

0 mark: Not attempted/wrong

50% marks: Partially attempted/correct

100% marks: Complete and correct



1: Advanced SQL, Triggers, and Indexing (10 Marks)

Q1(a) The dataset to be used in this exercise is the College_Admission database provided along with this assignment. The dataset contains a normalized schema for a table of students, colleges, and information about students applying to colleges. You are required to load the data in Oracle Database and use SQL Developer to answer the following questions. For each question please provide query statements and results. **(1*3=3 marks)**

- Retrieve students' names with majors for which they have applied.
- Retrieve the name, GPA, and application decision of students from the high school with less than 1000 students and have applied to major in CS at Stanford.
- Compute the amount by which the average GPA of students applying to CS exceeds the average GPA of students who are not applying to CS. *[Hint: compute two subqueries in the from clause, one of them computes the average GPA of CS applicants and one the average GPA of non CS applicants, in the select clause do the subtraction of the non-CS GPA from the CS-GPA].*

Q1(b) In the next task, use the same database (College_Admission) by using PL/SQL block to work on triggers for the scenario given below: **(4 marks)**

Create a row-level trigger on the student table that will fire after insertions into the student table by checking the GPA value of a new record. If the GPA of the inserted student is greater than 3.3, or less than or equal to 3.6, that student will be automatically applying to Stanford for a geology major and applying to MIT for a biology major. In other words, if the GPA of a new student satisfies the given condition ($GPA > 3.3$ or $GPA \leq 3.6$), two new rows should be automatically added to Apply table, one with values (sID of the new Student, cName: 'Stanford', major: 'geology', Decision: NULL, and second row with values (sID of the new student, cName: 'MIT', major: 'biology', Decision: NULL).

To test the above trigger, create and execute SQL statements to insert a new row in the student table with satisfying GPA conditions. E.g.

```
insert into Student values ('111', 'Kevin', 3.5, 1000);  
insert into Student values ('222', 'Lori', 3.8, 1000);
```

The first insert statement satisfies the GPA condition ($GPA > 3.3$ or $GPA \leq 3.6$), so two new tuples should be automatically added in the Apply table. You should provide trigger code, test SQL and query results showing new records in the student, and Apply tables according to your insert statement.

Q1(c) In the next task, you will be working on the same database to work on indexes by considering the following scenario: **(3 marks)**

Create a tree-based index (one at a time) on the following attribute pairs (Note: by default, indexes are tree-based).

```
Student.sID, College.cName
```



```
Student.sID, Student.GPA  
Apply.cName, College.cName  
Apply.sID, Student.GPA
```

Execute the following query each time you create an index, and evaluate which two pairs of the index are most useful for speeding up query execution. For each pair, discuss why or why not that index pair is useful based on the number of disc blocks required to access to retrieve a record.

```
Select * From Student, Apply, College  
Where Student.sID = Apply.sID and Apply.cName = College.cName  
And Student.GPA > 1.5 And College.cName < 'Cornell'
```

2: Transaction Management (10 Marks)

Q2(a): Consider a relation $R(A)$ containing $\{(5),(6)\}$ and two transactions:

T1: Update R set $A = A+1$;

T2: Update R set $A = 2*A$.

Suppose both transactions are submitted under the **isolation and atomicity** properties. Provide all possible final states of R (values of R) by considering all possible serializable and non-serializable schedules. **(2 marks)**

Q2(b) Consider a table $R(A)$ containing $\{(1),(2)\}$ and following two transactions run concurrently:

T1: Update R set $A = 2*A$; commit;

T2: Select avg(A) from R ; commit;

If transaction $T2$ executes using **READ UNCOMMITTED**, what are the possible average values of $T2$ returns? Provide an explanation for each value it returns. **(2 marks)**

Q2(c) Consider table $R(A)$ containing $\{(1),(2)\}$. Suppose we have the following two transactions running concurrently:

T1: Update T set $A=2*A$;
insert into R values (6);
Commit;

T2: Select avg(A) from R ;
Select avg(A) from R ;
Commit;

If transaction $T2$ executes using **REPEATABLE READ**, what are the possible values returned by $T2$ in its **SECOND** select statement? Provide an explanation for each value it returns. **(2 marks)**

Q2(d) For each of the following schedules of read, write, commit and abort actions done by transactions $T1, T2, T3$, state whether they are recoverable or not. If not recoverable, what type of inconsistencies might we have in the final database state? If recoverable, which of the other



transactions need to be rolled back? Note $r_i(A)$ and $w_i(A)$ mean that transaction T_i reads and writes database object A, respectively. (2*2=4 marks)

S1		
T1	T2	T3
w1(A);	r2(A); w2(A*2)	
w1(B);	r2(B);	r3(B); r3(B+5);
	commit;	commit;
abort;		

S2		
T1	T2	T3
r1(A); r1(B);	r2(A); w2(A*2)	
w1(B);		r3(B); w3(B+5);
abort;		

3: Querying XML Data (10 Marks)

Q3 In this task, you are required to use the Company XML schema provided along with the lab manual (Chapter 3). Specify the following views by writing and executing XQuery expressions on the company.xml document. For each query, provide query expression and results. (2*5=10 marks)

Note: You can use any XML editor using the software guide provided along with the lab manual.

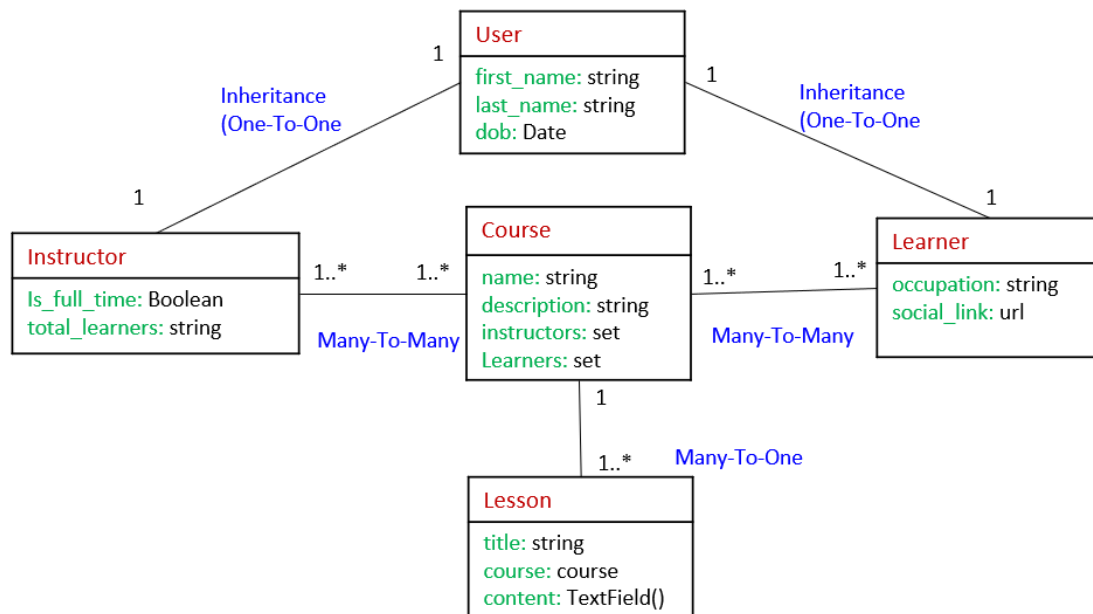
- Get distinct project numbers of projects in which employees work. The results should be ordered by project number.
- Create a view that has the department name, manager name, and manager salary for every department.
- A view that has the employee name, supervisor name, and employee salary for each employee who works in the 'Research' department.



- iv. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.
- v. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

4: Object-Relational Database (10 Marks)

Answer questions **4(a)**, **4(b)**, and **4(c)** by considering the following UML diagram for the online course application we have used in the lab for Object-Relational Mapping.



Q4(a) Note that Instructor and learner have Many-To-Many relationships with the course. We created class models for User, Instructor, Learner, Course, and Lesson to directly map these entities to objects. In addition, we created an Enrollment class to allow students to get enrolled for a course. We have added Instructors in the Course class in the lab. In this task, complete the below code by updating the Course class by adding Many-To-Many relationship with Learner via Enrollment relationship. **(2 marks)**



Update Course model

```
class Course(models.Model):  
    name = models.CharField(null=False, max_length=100, default='online course')  
    description = models.CharField(max_length=500)  
    # Many-To-Many relationship with Instructors  
    instructors = models.ManyToManyField(Instructor)  
  
    # Complete below code for Many-To-Many relationship with Learner via Enrollment relationship, similar to  
    # instructors above, however you have to lookup learner via enrollment class.  
  
    learners = "your code should go here"  
  
    def __str__(self):  
        return "Name: " + self.name + ", " + \  
            "Description: " + self.description
```

Q4(b) After defining class models in the lab assignment, we created objects by defining methods for User, Instructor, Learner, Course, and Lesson using the 'write.py' file. We also defined a method for creating objects for Course_Enrollment relationship. In this task, you are required to open the 'write.py' file and append a method for Course_instructor_relationship() to add courses for instructors. You are required to complete the below code. **(3 marks)**

```
def write_course_instructor_relationships():  
  
    # Get at least three instructors [Hint: use objects.get() method on the Instructor class]  
    # Your code should go here  
  
    # Get at least two related courses [Hint: use objects.get() method on the Course class]  
    # Your code should go here  
  
    # Add instructors to courses [hint: use Add() method on the Instructor class]  
    # Your code should go here  
  
    write_course_instructor_relationships()
```

Save the 'write.py' file and run it in the terminal to populate the 'course_instructor' table in the database. Provide a screenshot of course_instructor table with three rows referring to rows in the course and instructor tables.



Save the 'write.py' file and run it in the terminal to populate the 'course_instructor' table in the database. Provide a screenshot of course_instructor table with three rows referring to rows in the course and instructor table. Also, provide a screenshot of the course and user table to refer to rows in these tables corresponding to the given instructor of the course from the course_instructor table.

Q4(c) At this point, you have defined models and created objects. Now you will apply your skills of querying objects. In this task, you are required to create 'Read.py' file under your 'online_Courses' project directory.

Open the 'Read.py' file; first, copy Django-specific settings from the 'wsgi.py' file, set database connection, and import model classes, as we did in the lab session. Create code to perform the following queries: **(1*5 = 5 marks)**

- i. Find learners with last name "Smith".
- ii. Find two youngest learners (ordered by dob).
- iii. Retrieve all learners for the "Cloud Application Development with Database" course.
- iv. Retrieve instructors for "introduction to python" course.
- v. Retrieve occupation list of learners for the courses taught by instructor "Peter".

Format your code in a way that the query results should look like as given below. Provide a screenshot of your code and query results.



1. Find learners with last name `Smith`:

<QuerySet [

<Learner: First name: James, Last name: Smith, Date of Birth: 1982-07-16, Occupation: data_scientist, Social Link: <https://www.linkedin.com/james/>>,

<Learner: First name: Mary, Last name: Smith, Date of Birth: 1991-06-12, Occupation: dba, Social Link: <https://www.facebook.com/mary/>>,

<Learner: First name: David, Last name: Smith, Date of Birth: 1983-07-16, Occupation: developer, Social Link: <https://www.linkedin.com/david/>>,

<Learner: First name: John, Last name: Smith, Date of Birth: 1986-03-16, Occupation: developer, Social Link: <https://www.linkedin.com/john/>>]>

2. Find top two youngest learners:

<QuerySet [

<Learner: First name: Robert, Last name: Lee, Date of Birth: 1999-01-02, Occupation: student, Social Link: <https://www.facebook.com/robert/>>,

<Learner: First name: Mary, Last name: Smith, Date of Birth: 1991-06-12, Occupation: dba, Social Link: <https://www.facebook.com/mary/>>]

3. Retrieve all learners for "Cloud Application Development with Database" course

<QuerySet [

>/

Retrieve instructors for "introduction to python" course

<QuerySet [

>/

4. Retrieve instructors for "introduction to python" course

<QuerySet [

>/

5. Retrieve occupation list of learners for the courses taught by instructor "Peter"

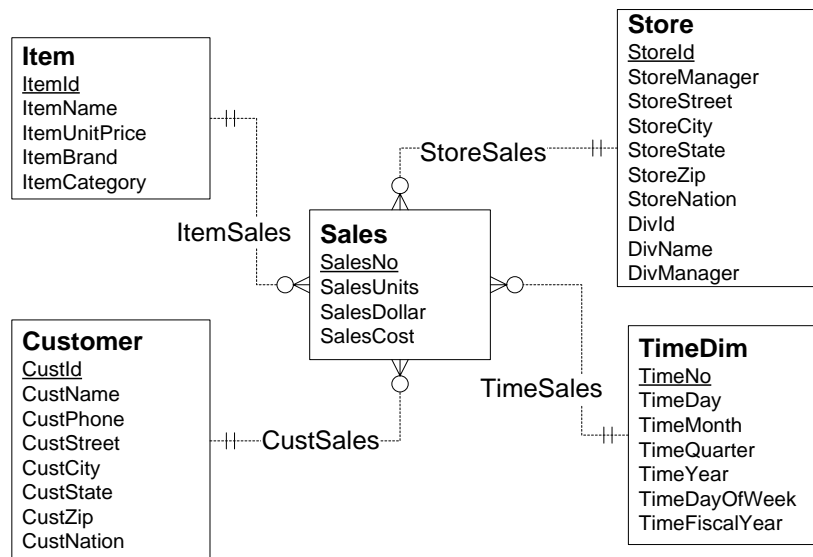
<QuerySet [

</



5: Data Warehousing and OLAP (10 Marks)

Q5 In this task, you will perform “Online Analytical Processing” (OLAP) style queries over a simple “star schema” given below. The corresponding schema, along with the database, is provided with this assignment. You are required to load the schema in Oracle Database and use SQL developer to perform the following queries: (2*5=10 marks)



- Summarize (sum, min, and count) store sales for USA and Canada in 2016 by store zip code and month without using subtotal operators. Provide query code and results.
- Use the same problem in part one and generate all possible subtotals by zip code and month using the data CUBE operator. Provide query code and results in your report.
- Summarize (SUM, COUNT, and MIN) store sales for USA and Canada between 2016 and 2017 by year and month. Generate partial subtotals for *year* and *month* using the ROLLUP operator. Provide query code and results in your report.
- Rewrite the query in part 3, and provide a set of subtotals on year, quarter, and month. Sort the results in a convenient order.
- Identify differences between results containing the GROUP BY clause (but not subtotal operators) and data cube and Rollup Results from queries in 1, 2, and 3. You should compare and contrast results based on the number of rows and subtotals groups. Explain why or why not each type of subtotal is useful for supporting business-specific queries.