COMP508 DATABASE SYSTEM DESIGN

SEMESTER 2, 2023

Paired Assignment

Database Project **50% of the final grade**

Due: 5:00 pm, 25th October 2023

Part	Tasks	Marks
Part A	Entity Relationship Modelling and Logical Database Design ➢ *Tasks 1 and 2*	35
Part B	Database Implementation ➢ *Tasks 3 and 4*	35
Part C	Construct SQL Queries ➢ *Task 5*	30

Student IDS:

Hayden Richard-Marsters – 21152003
Vishal Nirmalan - 22180968

1. Introduction to Assignment Requirements

This assignment is designed to provide students with practical exposure to the stages involved in database development, including the development of entity relationship modelling, database design and implementation. It provides an opportunity for students to apply to a larger case study the database concepts discussed in class and practised in exercises.

A separate document on Canvas: Assignments – 'Case Study for Database Project', describes the case study in detail. Please read the case study carefully. Students may ask for additional clarification of the case study on the discussion board on Canvas.

This is a paired assignment. Pairs of students within the same lab stream should complete tasks together. However, you have the option to work independently with the approval of the Course leader.

The details of the three parts of the assignment and the tasks are specified in Section 2. Section 3 describes the submission details and dates. The marking scheme is in Section 4.

Expectations

The assignment includes 3 parts. Part A consists of Tasks 1 and 2. Part B is comprised of Tasks 3 and 4. Part C has task 5. Students are required to attempt **all the tasks**.

Tools & Templates

You must use **Oracle SQL Developer** and the specified CASE tool, **Visual Paradigm CE** version for this assignment**.**

Plagiarism

Plagiarism means borrowing from the work of another without indicating by referencing that the ideas expressed are not one's own.

Unauthorised Collaboration

Unauthorised collaboration means joint effort between students or students and others in preparing material submitted for assessment, except where this has been pre-approved by the paper programme. Students are encouraged to discuss matters covered in classes, but the expression of ideas and arguments must be the student's work.

ACADEMIC INTEGRITY GUIDELINES

AUT takes Academic Integrity very seriously and you are reminded that t**he following actions may be deemed to constitute a breach of the General Academic Regulations Part 7: Academic Discipline, Section 2 Dishonesty During Assessment or Course of Study**

- 2.1.1 copies from, or inappropriately communicates with another person
- 2.1.3 plagiarises the work of another person without indicating that the work is not the student's own – using the full work or partial work of another person without giving due credit to the original creator of that work
- 2.1.4 collaborates with others in the preparation of material, except where this has been approved as an assessment requirement.
- 2.1.5 resubmits previously submitted work without prior approval of the assessment board
- 2.1.6 using any other unfair means.

ADDITIONAL INFORMATION

- Your assessment responses must be your own work. You may be required to orally defend your responses to assessment questions.

2. Assignment Parts and Tasks

Part A Entity Relationship Modelling and Logical Database Design (35 marks)

Task 1 Identify Entities & Develop Business Rules [10 marks]

Identify all entities and relationships in the case study and develop a set of **business rules**. You should follow the syntax given below. Two rules must describe each relationship, one in each direction.

Entities:

Roads

- *Road_ID*
- *Location_ID*
- *Road_Name*
- *Road_Description*
- *Road_Length*
- *Road_Main*
- *Road_SubSection*

Projects

- *Project_Code*
- *Project_Name*
- *Project_Description*
- *Project_DateStart*
- *Project_DateComplete*

Employees

- *Employee_ID*
- *Employee_FirstName*
- *Employee_LastName*
- *Employee_HireDate*
- *Employee_DOB*
- *Employee_Gender*
- *Employee_Postal*
- *Employee_PhoneNo*
- *Employee_Email*

Location

- *Location_ID*
- *Location_Name*
- *Location_Latitude*
- *Location_Longitude*
- *Location_Description*

Contracts

- *Contract_Number*
- *Contract_Name*
- *Contract_Description*
- *Contract_EstCost*
- *Contract_ActCost*
- *Contract_StartDate*
- *Contract_EndDate*

Roles

- *Role_Name*
- *Role_Description*
- *Role_StartDate*
- *Role_EndDate*

Contractor

- *Contractor_Name*
- *Contractor_Address*
- *Contractor_Contact*

Business rules:

Each ENTITY_1 *May/Must* Relationship_Verb_Phrase *number* ENTITY_2

ROADS/LOCATION (Starting location)

Each Road must start at one and only one location. A location may have many roads starting from it.

ROADS/LOCATION (Ending location)

Each Road must end at one and only one Location. A location may have many roads ending at it.

PROJECTS/ROAD

Each project must be associated with at least one Road and each road may be assigned to zero or many projects

ROAD/ROAD

A road may have many sub-roads, but a sub-road may only belong to one main road.

PROJECTS/ROLES

Each project must be assigned at least one primary role, and many roles may work on one project

ROLES/EMPLOYEE

A role may be assigned to zero or many AT employees, and each AT employee may be assigned zero or many roles.

EMPLOYEES/CONTRACTS

A contract may have one or contract managers (AT employees) and one or many AT employees may manage a contract.

CONTRACTS/CONTRACTOR

A contractor may work on zero or many contracts. A contract must be assigned only one project

CONTRACTS/PROJECTS

A project may have one or many contracts. Each contract may only be assigned to one project.

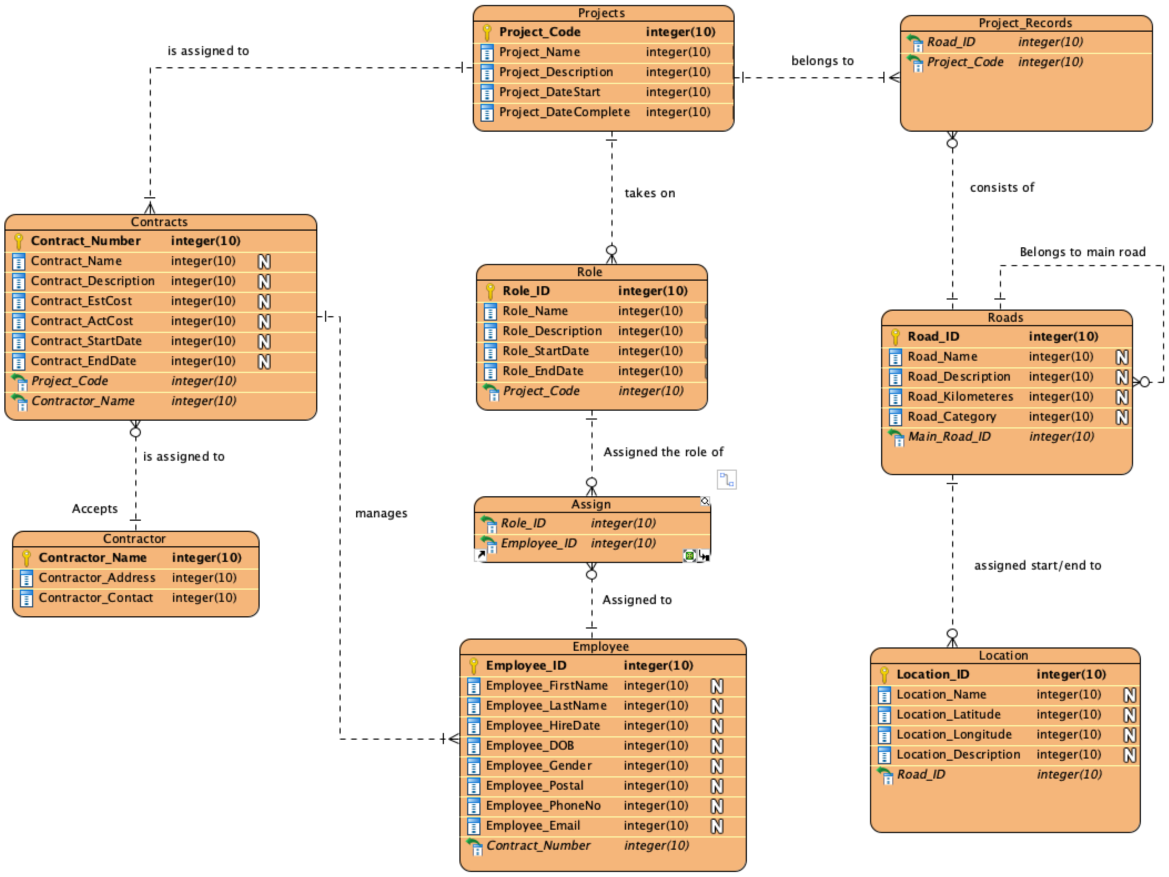
Task 2 Construct a logical Entity Relationship Diagram (ERD) [25 marks]

Based on the business rules developed in Task 1, construct a logical ERD for the case study using the Visual Paradigm tool.

- Identify all attributes in each entity, including all primary and foreign keys (transform any composite and multivalued attributes according to the rules of the relational model).
- Name all relationships (i.e. you must use verb phrases on both sides of each relationship)
- Identify the Cardinality and Participation for each relationship.
- Resolve all many-to-many (M:N) relationships.
- Describe any assumptions you have made in a text note on the diagram.

Please note:

1) You don't need to identify data types in your ERD.
2) You don't need to answer tasks 1 and 2 questions separately. You identify the attributes, entities and relationships on the diagram.


is assigned to
Projects
Project_Code integer(10)
Project_Name integer(10)
Project_Description integer(10)
Project_DateStart integer(10)
Project_DateComplete integer(10)
belongs to
Project_Records
Road_ID integer(10)
Project_Code integer(10)
takes on
consists of
Contracts
Contract_Number integer(10)
Contract_Name integer(10)
Contract_Description integer(10)
Contract_EstCost integer(10)
Contract_ActCost integer(10)
Contract_StartDate integer(10)
Contract_EndDate integer(10)
Project_Code integer(10)
Contractor_Name integer(10)
Role
Role_ID integer(10)
Role_Name integer(10)
Role_Description integer(10)
Role_StartDate integer(10)
Role_EndDate integer(10)
Project_Code integer(10)
Belongs to main road
Roads
Road_ID integer(10)
Road_Name integer(10)
Road_Description integer(10)
Road_Kilometeres integer(10)
Road_Category integer(10)
Main_Road_ID integer(10)
is assigned to
Assigned the role of
Accepts
manages
Assign
Role_ID integer(10)
Employee_ID integer(10)
Contractor
Contractor_Name integer(10)
Contractor_Address integer(10)
Contractor_Contact integer(10)
assigned start/end to
Assigned to
Employee
Employee_ID integer(10)
Employee_FirstName integer(10)
Employee_LastName integer(10)
Employee_HireDate integer(10)
Employee_DOB integer(10)
Employee_Gender integer(10)
Employee_Postal integer(10)
Employee_PhoneNo integer(10)
Employee_Email integer(10)
Contract_Number integer(10)
Location
Location_ID integer(10)
Location_Name integer(10)
Location_Latitude integer(10)
Location_Longitude integer(10)
Location_Description integer(10)
Road_ID integer(10)

Part B Database Implementation (35 marks)

In this part, you should develop a database based on your logical ERD developed in Part A. You must populate the tables with realistic sample data.

Task 3 Create tables [25 marks]

Create the tables in Oracle.[1] Write SQL scripts defining each table. The table definitions should include

- All attributes with appropriate data types
- All appropriate constraints, such as primary key, foreign keys, and check statements (CHECK constraints).
- All constraints **must** be given names. Naming standards must be used.

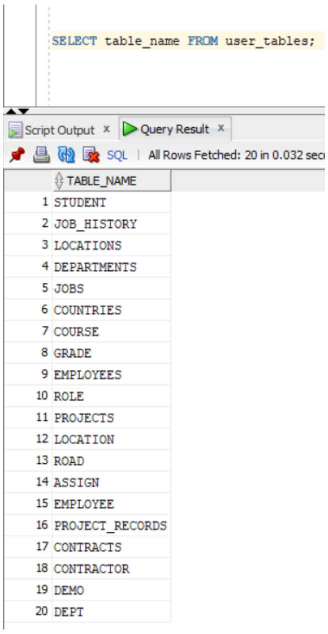
Task 4 Populate data [10 marks]

Insert a small sample of realistic test data (5 –10 rows minimum) into each table. All test data must be meaningful to demonstrate your understanding of the data.

[1] You must use Oracle SQL Developer for Parts B and C.

After creating all your tables (Task 3) and inserting data (Task 4), run the SQL SELECT statement below.

SELECT table_name FROM user_tables;



```
SELECT table_name FROM user_tables;
```

All Rows Fetched: 20 in 0.032 seco

```
    TABLE_NAME
 1  STUDENT
 2  JOB_HISTORY
 3  LOCATIONS
 4  DEPARTMENTS
 5  JOBS
 6  COUNTRIES
 7  COURSE
 8  GRADE
 9  EMPLOYEES
10  ROLE
11  PROJECTS
12  LOCATION
13  ROAD
14  ASSIGN
15  EMPLOYEE
16  PROJECT_RECORDS
17  CONTRACTS
18  CONTRACTOR
19  DEMO
20  DEPT
```

The statement will retrieve a list of all tables in your Oracle account (i.e., all tables you created this semester). Include the results of the SELECT statement in your assignment.

You must also include the following for each table in your database:

- SQL table creation script
- SQL insert script (used for inserting sample data)
  - Evidence of Sample data - you must run "SELECT * FROM *table_name"* (where *table_name* is the table's name) for each table in Task 3 and include results of SELECT statements for each table.

QUERY SCRIPT (ALL INCLUDED)

```
CREATE TABLE Projects (
   Project_Code VARCHAR2(5) PRIMARY KEY,
   Project_Name VARCHAR2(50),
   Project_Description VARCHAR2(250),
   Project_DateStart DATE,
   Project_DateComplete DATE
);

CREATE TABLE Role (
   Role_ID VARCHAR2(5) PRIMARY KEY,
   Role_Name VARCHAR2(50),
   Role_Description VARCHAR2(250),
   Role_StartDate DATE,
   Role_EndDate DATE
);

CREATE TABLE Employee (
   Employee_ID VARCHAR2(5) PRIMARY KEY,
   Employee_FirstName VARCHAR2(15),
   Employee_LastName VARCHAR2(15),
   Employee_HireDate DATE,
   Employee_DOB DATE,
   Employee_Gender VARCHAR2(10),
   Employee_Postal VARCHAR2(150),
   Employee_PhoneNo NUMBER,
   Employee_Email VARCHAR2(30)
);

CREATE TABLE Road (
   Road_ID VARCHAR2(5) PRIMARY KEY,
   Road_Name VARCHAR2(30),
   Road_Description VARCHAR(250),
   Road_Kilometers NUMBER,
   Road_Category VARCHAR2(15)
);

CREATE TABLE Location (
   Location_ID VARCHAR2(5) PRIMARY KEY,
   Location_Name VARCHAR2(50),
   Location_Latitude NUMBER,
   Location_Longitude NUMBER,
   Location_Description VARCHAR2(250)
);

CREATE TABLE Contracts (
   Contract_Number NUMBER PRIMARY KEY,
   Contract_Name VARCHAR2(50),
   Contract_Description VARCHAR2(250),
   Contract_EstCost NUMBER,
   Contract_ActCost NUMBER,
   Contract_StartDate DATE,
   Contract_EndDate Date
);

CREATE TABLE Contractor (
   Contractor_Name VARCHAR2(50) PRIMARY KEY,
   Contractor_Address VARCHAR2(150),
   Contractor_Contact VARCHAR2(20)
);

CREATE TABLE Project_Records (
   Road_ID VARCHAR2(5),
   FOREIGN KEY (Road_ID) REFERENCES Road(Road_ID),
   Project_Code VARCHAR2(5),
   FOREIGN KEY (Project_Code) REFERENCES Projects(Project_Code)
);

CREATE TABLE Assign (
   Employee_ID VARCHAR2(5),
   FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_id),
   Role_ID VARCHAR2(5),
   FOREIGN KEY (Role_ID) REFERENCES Role(Role_ID)
);

--------------------------------------------------------------------------------

ALTER TABLE Role
ADD Project_Code VARCHAR2(5);  -- Add the column first

ALTER TABLE Role
ADD CONSTRAINT fk_Project_Role
FOREIGN KEY (Project_Code)
REFERENCES Projects(Project_Code);  -- Add the constraint separately


ALTER TABLE Employee
ADD Contract_Number NUMBER;  -- Add the column first

ALTER TABLE Employee
ADD CONSTRAINT fk_Contract_Employee
FOREIGN KEY (Contract_Number)
REFERENCES Contracts(Contract_Number);  -- Add the constraint with the correct reference

ALTER TABLE Road
```

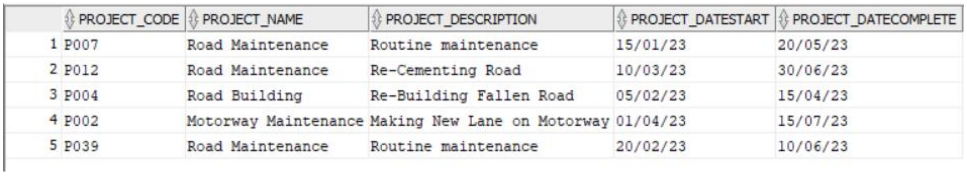
```
ADD Road_Main_Id VARCHAR2(5);

ALTER TABLE Road
ADD CONSTRAINT fk_Road_Main
FOREIGN KEY (Road_Main_Id)
REFERENCES Road(Road_ID);  -- Add the constraint with the correct reference

ALTER TABLE Location
ADD Road_ID VARCHAR2(5);

ALTER TABLE Location
ADD CONSTRAINT fk_Road_Location
FOREIGN KEY (Road_ID)
REFERENCES Road(Road_ID);  -- Corrected column names and references

ALTER TABLE Contracts
ADD Project_Code VARCHAR2(5);

ALTER TABLE Contracts
ADD CONSTRAINT fk_Project_Contracts
FOREIGN KEY (Project_Code)
REFERENCES Projects(Project_Code);  -- Corrected column names and references

ALTER TABLE Contracts
ADD Contractor_Name VARCHAR2(50);  -- Corrected column size

ALTER TABLE Contracts
ADD CONSTRAINT fk_Contractor_Contracts
FOREIGN KEY (Contractor_Name)
REFERENCES Contractor(Contractor_Name);  -- Corrected column names and references


 ---------------------------------------------------------------------

 INSERT INTO Projects (Project_Code, Project_Name, Project_Description,
Project_DateStart, Project_DateComplete)
VALUES ('P007', 'Road Maintenance', 'Routine maintenance',
TO_DATE('2023-01-15', 'YYYY-MM-DD'), TO_DATE('2023-05-20', 'YYYY-MM-DD'));
INSERT INTO Projects (Project_Code, Project_Name, Project_Description,
Project_DateStart, Project_DateComplete)
VALUES ('P012', 'Road Maintenance', 'Re-Cementing Road',
TO_DATE('2023-03-10', 'YYYY-MM-DD'), TO_DATE('2023-06-30', 'YYYY-MM-DD'));
INSERT INTO Projects (Project_Code, Project_Name, Project_Description,
Project_DateStart, Project_DateComplete)
VALUES ('P004', 'Road Building', 'Re-Building Fallen Road',
TO_DATE('2023-02-05', 'YYYY-MM-DD'), TO_DATE('2023-04-15', 'YYYY-MM-DD'));
INSERT INTO Projects (Project_Code, Project_Name, Project_Description,
Project_DateStart, Project_DateComplete)
VALUES ('P002', 'Motorway Maintenance', 'Making New Lane on Motorway',
TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-07-15', 'YYYY-MM-DD'));
INSERT INTO Projects (Project_Code, Project_Name, Project_Description,
Project_DateStart, Project_DateComplete)
VALUES ('P039', 'Road Maintenance', 'Routine maintenance',
TO_DATE('2023-02-20', 'YYYY-MM-DD'), TO_DATE('2023-06-10', 'YYYY-MM-DD'));

--

INSERT INTO Role (Role_ID, Role_Name, Role_Description, Role_StartDate,
Role_EndDate, Project_Code)
VALUES ('R007', 'Construction Manager', 'Oversees and manages construction staff',
TO_DATE('2023-01-15', 'YYYY-MM-DD'), TO_DATE('2023-05-20', 'YYYY-MM-DD'), 'P007');
INSERT INTO Role (Role_ID, Role_Name, Role_Description, Role_StartDate,
Role_EndDate, Project_Code)
VALUES ('R002', 'Construction Member', 'Part of construction staff',
TO_DATE('2023-03-10', 'YYYY-MM-DD'), TO_DATE('2023-06-30', 'YYYY-MM-DD'), 'P012');
INSERT INTO Role (Role_ID, Role_Name, Role_Description, Role_StartDate,
Role_EndDate, Project_Code)
VALUES ('R003', 'Temporary Traffic Manager', 'Oversees redirecting traffic',
TO_DATE('2023-02-05', 'YYYY-MM-DD'), TO_DATE('2023-04-15', 'YYYY-MM-DD'), 'P004');
INSERT INTO Role (Role_ID, Role_Name, Role_Description, Role_StartDate,
Role_EndDate, Project_Code)
```

```
VALUES ('R047', 'Cement Truck Driver', 'Operates cement truck',
TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-07-15', 'YYYY-MM-DD'), 'P002');
INSERT INTO Role (Role_ID, Role_Name, Role_Description, Role_StartDate,
Role_EndDate, Project_Code)
VALUES ('R012', 'Crane Operator', 'Operates cranes',
TO_DATE('2023-02-20', 'YYYY-MM-DD'), TO_DATE('2023-06-10', 'YYYY-MM-DD'), 'P039');

--


INSERT INTO Contractor (Contractor_Name, Contractor_Address, Contractor_Contact)
VALUES ('StrongBuild Constructors', '101 Elmwood Avenue', '0800 333 999');
INSERT INTO Contractor (Contractor_Name, Contractor_Address, Contractor_Contact)
VALUES ('Apex Builders Group', '2 Riverfront Drive', '0800 837 432');
INSERT INTO Contractor (Contractor_Name, Contractor_Address, Contractor_Contact)
VALUES ('Heritage Construction', '39 Sunflower Street', '0800 544 433');
INSERT INTO Contractor (Contractor_Name, Contractor_Address, Contractor_Contact)
VALUES ('Skyline Builders Inc.', '13 Willowbrook Road', '027 348 331');
INSERT INTO Contractor (Contractor_Name, Contractor_Address, Contractor_Contact)
VALUES ('Precision Constructors LLC', '149 Meadowbrook Lane', '0800 132 465');

--


INSERT INTO Contracts (Contract_Number, Contract_Name, Contract_Description,
Contract_EstCost, Contract_ActCost, Contract_StartDate, Contract_EndDate, Project_Code, Contractor_Name)
VALUES ('1', 'Re-Cementing Contract', 'Re-Cementing of road', '12500', '14000',
TO_DATE('2023-03-10', 'YYYY-MM-DD'), TO_DATE('2023-06-30', 'YYYY-MM-DD'), 'P012', 'StrongBuild Constructors');
INSERT INTO Contracts (Contract_Number, Contract_Name, Contract_Description,
Contract_EstCost, Contract_ActCost, Contract_StartDate, Contract_EndDate, Project_Code, Contractor_Name)
VALUES ('2', 'Motorway Lane Contract', 'Adding new lane to motorway', '1500000', '1300000',
TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-07-15', 'YYYY-MM-DD'), 'P002', 'StrongBuild Constructors');
INSERT INTO Contracts (Contract_Number, Contract_Name, Contract_Description,
Contract_EstCost, Contract_ActCost, Contract_StartDate, Contract_EndDate, Project_Code, Contractor_Name)
VALUES ('3', 'Motorway Lane Contract', 'Adding new lane to motorway', '1500000', '1300000',
TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-07-15', 'YYYY-MM-DD'), 'P002', 'StrongBuild Constructors');
INSERT INTO Contracts (Contract_Number, Contract_Name, Contract_Description,
Contract_EstCost, Contract_ActCost, Contract_StartDate, Contract_EndDate, Project_Code, Contractor_Name)
VALUES ('4', 'Road Re-Build Contract', 'Rebuilding fallen road', '2000000', '3500000',
TO_DATE('2023-02-05', 'YYYY-MM-DD'), TO_DATE('2023-04-15', 'YYYY-MM-DD'), 'P004', 'Apex Builders Group');
INSERT INTO Contracts (Contract_Number, Contract_Name, Contract_Description,
Contract_EstCost, Contract_ActCost, Contract_StartDate, Contract_EndDate, Project_Code, Contractor_Name)
VALUES ('5', 'Road Maintenance Contract', 'Routine maintenance', '8000', '12000',
TO_DATE('2023-01-15', 'YYYY-MM-DD'), TO_DATE('2023-05-20', 'YYYY-MM-DD'), 'P007', 'Heritage Construction');

--

INSERT INTO Employee (Employee_ID, Employee_FirstName, Employee_LastName, Employee_HireDate, Employee_DOB,
Employee_Gender, Employee_Postal, Employee_PhoneNo, Employee_Email)
VALUES ('E0001', 'Emily', 'Mitchell', TO_DATE('2019-01-15', 'YYYY-MM-DD'), TO_DATE('2001-05-20', 'YYYY-MM-DD'), 'Female', '23
Maplewood Lane', '0223922302', 'emilymitchell@at.co.nz');

INSERT INTO Employee (Employee_ID, Employee_FirstName, Employee_LastName, Employee_HireDate, Employee_DOB,
Employee_Gender, Employee_Postal, Employee_PhoneNo, Employee_Email)
VALUES ('E0002', 'Alexander', 'Johnson', TO_DATE('2017-07-25', 'YYYY-MM-DD'), TO_DATE('1998-06-20', 'YYYY-MM-DD'), 'Male', '47
Sunset Boulevard', '0210745353', 'alexanderjohnson@at.co.nz');

INSERT INTO Employee (Employee_ID, Employee_FirstName, Employee_LastName, Employee_HireDate, Employee_DOB,
Employee_Gender, Employee_Postal, Employee_PhoneNo, Employee_Email)
VALUES ('E0003', 'Sophia', 'Williams', TO_DATE('2021-04-13', 'YYYY-MM-DD'), TO_DATE('2003-11-11', 'YYYY-MM-DD'), 'Female', '19
Pinecrest Avenue', '0274346847', 'sophiawilliams@at.co.nz');

INSERT INTO Employee (Employee_ID, Employee_FirstName, Employee_LastName, Employee_HireDate, Employee_DOB,
Employee_Gender, Employee_Postal, Employee_PhoneNo, Employee_Email)
VALUES ('E0004', 'Benjamin', 'Davis', TO_DATE('2020-07-07', 'YYYY-MM-DD'), TO_DATE('1983-06-21', 'YYYY-MM-DD'), 'Male', '103
Oakridge Drive', '0214392312', 'benjamindavis@at.co.nz');

INSERT INTO Employee (Employee_ID, Employee_FirstName, Employee_LastName, Employee_HireDate, Employee_DOB,
Employee_Gender, Employee_Postal, Employee_PhoneNo, Employee_Email)
```

```
VALUES ('E0005', 'Olivia', 'Martinez', TO_DATE('2019-11-05', 'YYYY-MM-DD'), TO_DATE('1997-11-20', 'YYYY-MM-DD'), 'Female', '3
Meadowbrook Road', '0210832233', 'oliviamartinez@at.co.nz');


--

INSERT INTO Road (Road_ID, Road_Name, Road_Description, Road_Kilometers, Road_Category)
VALUES ('RD001', 'Elmwood Avenue', 'Long road in suburban area', '4.1', 'Main Road');
INSERT INTO Road (Road_ID, Road_Name, Road_Description, Road_Kilometers, Road_Category)
VALUES ('RD003', 'Sunset Boulevard', 'Long road near beach', '3.7', 'Main Road');
INSERT INTO Road (Road_ID, Road_Name, Road_Description, Road_Kilometers, Road_Category)
VALUES ('RD004', 'Maplewood Lane', 'Long road in city', '2.2', 'Main Road');
INSERT INTO Road (Road_ID, Road_Name, Road_Description, Road_Kilometers, Road_Category, Road_Main_ID)
VALUES ('RD005', 'Oakridge Drive', 'Small road near beach', '0.3', 'Sub Road', 'RD003');
INSERT INTO Road (Road_ID, Road_Name, Road_Description, Road_Kilometers, Road_Category, Road_Main_ID)
VALUES ('RD002', 'Meadowbrook Road', 'Small road in suburban area', '0.8', 'Sub Road', 'RD001');

--

INSERT INTO Location (Location_ID, Location_Name, Location_Latitude,
Location_Longitude, Location_Description, Road_ID)
VALUES ('L0001', 'Elmwood Avenue', '0.12324875', '0.3463521', 'East end of Elmwood Avenue', 'RD001');
INSERT INTO Location (Location_ID, Location_Name, Location_Latitude,
Location_Longitude, Location_Description, Road_ID)
VALUES ('L0002', 'Elmwood Avenue', '0.12324875', '0.55463521', 'West end of Elmwood Avenue', 'RD001');
INSERT INTO Location (Location_ID, Location_Name, Location_Latitude,
Location_Longitude, Location_Description, Road_ID)
VALUES ('L0003', 'Sunset Boulevard', '0.543125', '0.2345245', 'East end of Sunset Boulevard', 'RD003');
INSERT INTO Location (Location_ID, Location_Name, Location_Latitude,
Location_Longitude, Location_Description, Road_ID)
VALUES ('L0004', 'Maplewood Lane', '0.6745744', '0.346722', 'North end of Maplewood Lane', 'RD004');
INSERT INTO Location (Location_ID, Location_Name, Location_Latitude,
Location_Longitude, Location_Description, Road_ID)
VALUES ('L0005', 'Oakridge Drive', '0.7894232', '0.3456745', 'South end of Oakridge Drive', 'RD005');

--

INSERT INTO Project_Records (Road_ID, Project_Code)
VALUES ('RD001', 'P007');
INSERT INTO Project_Records (Road_ID, Project_Code)
VALUES ('RD002', 'P012');
INSERT INTO Project_Records (Road_ID, Project_Code)
VALUES ('RD003', 'P004');
INSERT INTO Project_Records (Road_ID, Project_Code)
VALUES ('RD004', 'P002');
INSERT INTO Project_Records (Road_ID, Project_Code)
VALUES ('RD005', 'P039');

--

INSERT INTO Assign (Employee_ID, Role_ID)
VALUES ('E0001', 'R007');
INSERT INTO Assign (Employee_ID, Role_ID)
VALUES ('E0002', 'R002');
INSERT INTO Assign (Employee_ID, Role_ID)
VALUES ('E0003', 'R003');
INSERT INTO Assign (Employee_ID, Role_ID)
VALUES ('E0004', 'R047');
INSERT INTO Assign (Employee_ID, Role_ID)
VALUES ('E0005', 'R012');

COMMIT;

----------------------------------------------------------------------
SELECT * FROM CONTRACTOR;
SELECT * FROM ROAD;
SELECT * FROM EMPLOYEE;
SELECT * FROM projects;
SELECT * FROM project_records;
SELECT * FROM contracts;
```

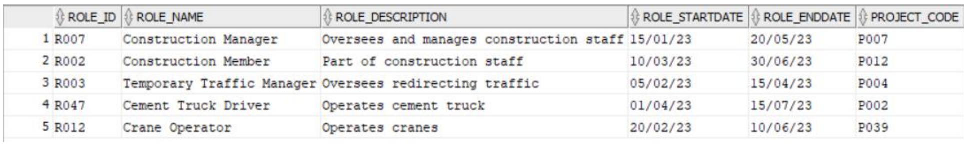
SELECT * FROM ASSIGN;
SELECT * FROM LOCATION;
SELECT * FROM ROLE;

Projects:



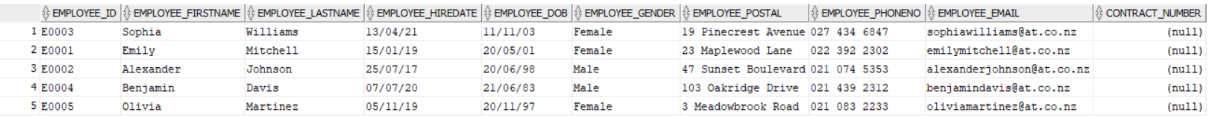
	PROJECT_CODE	PROJECT_NAME	PROJECT_DESCRIPTION	PROJECT_DATESTART	PROJECT_DATECOMPLETE
1	P007	Road Maintenance	Routine maintenance	15/01/23	20/05/23
2	P012	Road Maintenance	Re-Cementing Road	10/03/23	30/06/23
3	P004	Road Building	Re-Building Fallen Road	05/02/23	15/04/23
4	P002	Motorway Maintenance	Making New Lane on Motorway	01/04/23	15/07/23
5	P039	Road Maintenance	Routine maintenance	20/02/23	10/06/23

Role:



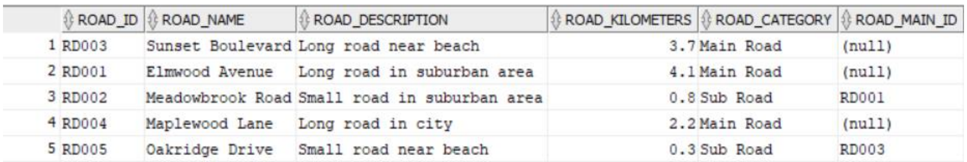
	ROLE_ID	ROLE_NAME	ROLE_DESCRIPTION	ROLE_STARTDATE	ROLE_ENDDATE	PROJECT_CODE
1	R007	Construction Manager	Oversees and manages construction staff	15/01/23	20/05/23	P007
2	R002	Construction Member	Part of construction staff	10/03/23	30/06/23	P012
3	R003	Temporary Traffic Manager	Oversees redirecting traffic	05/02/23	15/04/23	P004
4	R047	Cement Truck Driver	Operates cement truck	01/04/23	15/07/23	P002
5	R012	Crane Operator	Operates cranes	20/02/23	10/06/23	P039

Employee:



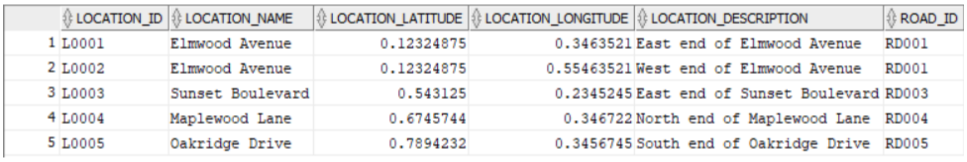
	EMPLOYEE_ID	EMPLOYEE_FIRSTNAME	EMPLOYEE_LASTNAME	EMPLOYEE_HIREDATE	EMPLOYEE_DOB	EMPLOYEE_GENDER	EMPLOYEE_POSTAL	EMPLOYEE_PHONENO	EMPLOYEE_EMAIL	CONTRACT_NUMBER
1	E0003	Sophia	Williams	13/04/21	11/11/03	Female	19 Pinecrest Avenue	027 434 6847	sophiawilliams@at.co.nz	(null)
2	E0001	Emily	Mitchell	15/01/19	20/05/01	Female	23 Maplewood Lane	022 392 2302	emilymitchell@at.co.nz	(null)
3	E0002	Alexander	Johnson	25/07/17	20/06/98	Male	47 Sunset Boulevard	021 074 5353	alexanderjohnson@at.co.nz	(null)
4	E0004	Benjamin	Davis	07/07/20	21/06/83	Male	103 Oakridge Drive	021 439 2312	benjamindavis@at.co.nz	(null)
5	E0005	Olivia	Martinez	05/11/19	20/11/97	Female	3 Meadowbrook Road	021 083 2233	oliviamartinez@at.co.nz	(null)

Road:



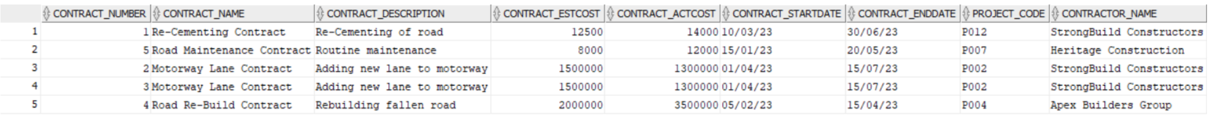
	ROAD_ID	ROAD_NAME	ROAD_DESCRIPTION	ROAD_KILOMETERS	ROAD_CATEGORY	ROAD_MAIN_ID
1	RD003	Sunset Boulevard	Long road near beach	3.7	Main Road	(null)
2	RD001	Elmwood Avenue	Long road in suburban area	4.1	Main Road	(null)
3	RD002	Meadowbrook Road	Small road in suburban area	0.8	Sub Road	RD001
4	RD004	Maplewood Lane	Long road in city	2.2	Main Road	(null)
5	RD005	Oakridge Drive	Small road near beach	0.3	Sub Road	RD003

Location:



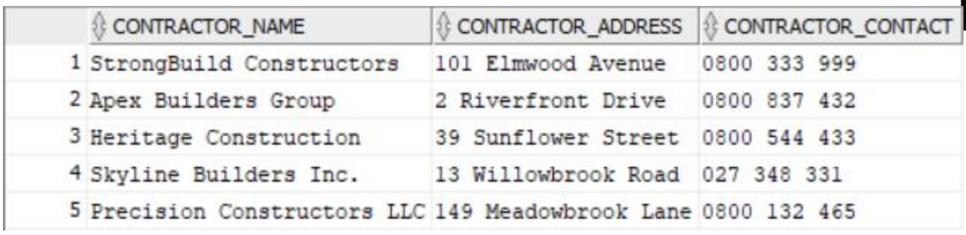
	LOCATION_ID	LOCATION_NAME	LOCATION_LATITUDE	LOCATION_LONGITUDE	LOCATION_DESCRIPTION	ROAD_ID
1	L0001	Elmwood Avenue	0.12324875	0.3463521	East end of Elmwood Avenue	RD001
2	L0002	Elmwood Avenue	0.12324875	0.55463521	West end of Elmwood Avenue	RD001
3	L0003	Sunset Boulevard	0.543125	0.2345245	East end of Sunset Boulevard	RD003
4	L0004	Maplewood Lane	0.6745744	0.346722	North end of Maplewood Lane	RD004
5	L0005	Oakridge Drive	0.7894232	0.3456745	South end of Oakridge Drive	RD005

Contracts:



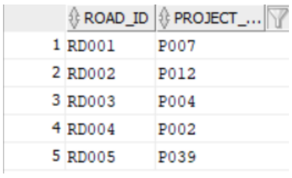
	CONTRACT_NUMBER	CONTRACT_NAME	CONTRACT_DESCRIPTION	CONTRACT_ESTCOST	CONTRACT_ACTCOST	CONTRACT_STARTDATE	CONTRACT_ENDDATE	PROJECT_CODE	CONTRACTOR_NAME
1	1	Re-Cementing Contract	Re-Cementing of road	12500	14000	10/03/23	30/06/23	P012	StrongBuild Constructors
2	5	Road Maintenance Contract	Routine maintenance	8000	12000	15/01/23	20/05/23	P007	Heritage Construction
3	2	Motorway Lane Contract	Adding new lane to motorway	1500000	1300000	01/04/23	15/07/23	P002	StrongBuild Constructors
4	3	Motorway Lane Contract	Adding new lane to motorway	1500000	1300000	01/04/23	15/07/23	P002	StrongBuild Constructors
5	4	Road Re-Build Contract	Rebuilding fallen road	2000000	3500000	05/02/23	15/04/23	P004	Apex Builders Group

Contractor:



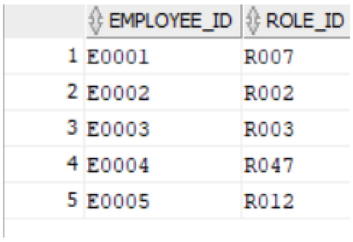
	CONTRACTOR_NAME	CONTRACTOR_ADDRESS	CONTRACTOR_CONTACT
1	StrongBuild Constructors	101 Elmwood Avenue	0800 333 999
2	Apex Builders Group	2 Riverfront Drive	0800 837 432
3	Heritage Construction	39 Sunflower Street	0800 544 433
4	Skyline Builders Inc.	13 Willowbrook Road	027 348 331
5	Precision Constructors LLC	149 Meadowbrook Lane	0800 132 465

Project Records:



	ROAD_ID	PROJECT_...
1	RD001	P007
2	RD002	P012
3	RD003	P004
4	RD004	P002
5	RD005	P039

Assign:



	EMPLOYEE_ID	ROLE_ID
1	E0001	R007
2	E0002	R002
3	E0003	R003
4	E0004	R047
5	E0005	R012

Part C Construct SQL Queries (30 marks)

Task 5 Construct SQL Queries [30 marks]

Identify **five** data retrieval requirements that would be used by the business described in the case study. Construct a SQL query to display the data for each requirement you have identified. So, you must construct five SQL queries in total. The queries must include a combination of the following SQL features:

- Use of restriction (WHERE clause), Arithmetic expressions, Concatenation of columns
- Use of comparison and logical operators
- Multiple tables join (e.g., Equijoin, Outer Join, and Self-join)

- Group functions (e.g., COUNT, SUM, AVG, MAX, MIN) and clauses (e.g., ORDER BY, GROUP BY, HAVING)

Note:

1. You may combine the preceding features in a single query. The design of each query should demonstrate your knowledge and application of SQL content covered in the course and your understanding of the correct usage of the various features and clauses of SQL. A simple ‘**select * from table_name**’ will fetch you a mark of zero in this task.

2. There should be **no more than one query that involves a single table.**

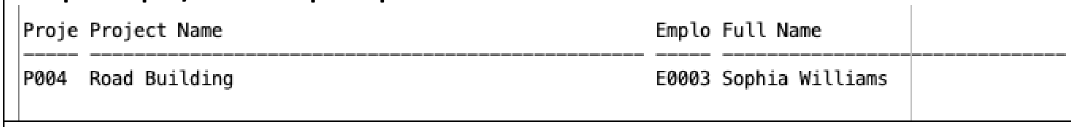
3. An example of a query is given on the next page.

One) Purpose of Query: Allow user input (&Substitution variable) of project code and return the name of the project as well as all employees assigned to that project, displaying employee details of their Full name (Concatenation of first and last name) and their employee ID

Query:

```
ACCEPT project_code CHAR PROMPT 'Enter Project Code: '

SELECT
   p.Project_Code AS "Project Code",
   p.Project_Name AS "Project Name",
   e.Employee_ID AS "Employee ID",
   e.Employee_FirstName ||' '|| e.Employee_LastName AS "Full Name"
FROM
   Projects p
JOIN
   Project_Records pr ON p.Project_Code = pr.Project_Code
JOIN
   Role r ON pr.Project_Code = r.Project_Code
JOIN
   Assign a ON r.Role_ID = a.Role_ID
JOIN
   Employee e ON a.Employee_ID = e.Employee_ID
WHERE
   p.Project_Code = UPPER('&project_code');
```

Script Output/Result: Input = p004



```
Proje Project Name                                          Emplo Full Name
----- ----------------------------------------------------- ----- ----------------------------------
P004  Road Building                                         E0003 Sophia Williams
```

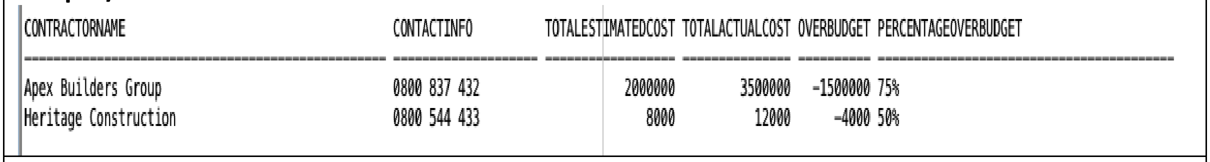
Two) Purpose of Query: Retrieve all contractors AT Transport contracts work to, but only the contractors who have gone over budget, displaying their total estimated and actual costs associated with all of their covered contracts giving their net budget value, as well as the percent amount over budget (amount over budget / estimated costs) Ordering them by most over budget to least by percent amount.

QUERY:

```
SELECT
   c.Contractor_Name AS ContractorName,
   c.Contractor_Contact AS ContactInfo,
   SUM(co.CONTRACT_ESTCOST) AS TotalEstimatedCost,
   SUM(co.CONTRACT_ACTCOST) AS TotalActualCost,
   SUM(co.CONTRACT_ESTCOST) - SUM(co.CONTRACT_ACTCOST) AS Overbudget,
   (SUM(co.CONTRACT_ACTCOST) - SUM(co.CONTRACT_ESTCOST)) / SUM(co.CONTRACT_ESTCOST) * 100 || '%' AS
PercentageOverBudget
FROM Contractor c
JOIN Contracts co ON c.Contractor_Name = co.Contractor_Name
```

```
GROUP BY c.Contractor_Name, c.Contractor_Contact
HAVING SUM(co.CONTRACT_ACTCOST) > SUM(co.CONTRACT_ESTCOST)
ORDER BY PercentageOverBudget DESC;
```

Output/Result



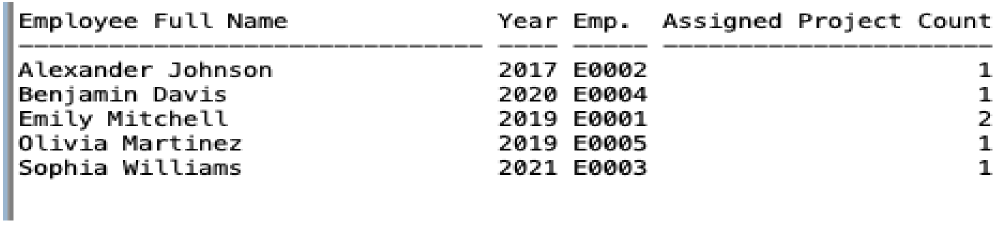
```
CONTRACTORNAME                                    CONTACTINFO        TOTALESTIMATEDCOST TOTALACTUALCOST OVERBUDGET PERCENTAGEOVERBUDGET
------------------------------------------------- ------------------ ------------------ --------------- ---------- ----------------------------------------
Apex Builders Group                               0800 837 432                  2000000         3500000   -1500000 75%
Heritage Construction                             0800 544 433                     8000           12000      -4000 50%
```

Three) Purpose of Query: Retrieve all employees a sorted by name list of all employees with the details of the Year they were hired (Derived from HireDate), Employee ID as well as the total number of projects assigned to them.

QUERY:

```
SELECT
   Employee_FirstName || ' ' || Employee_LastName AS "Employee Full Name",
   TO_CHAR(Employee_HireDate, 'YYYY') AS "Year Of Hire",
   Employee_ID AS "Emp. ID",
   (SELECT COUNT(*) FROM ASSIGN a WHERE a.Employee_ID = e.Employee_ID) AS "Assigned Project Count"
FROM
   Employee e
ORDER BY
   "Employee Full Name" ASC;
```

Output/Result:



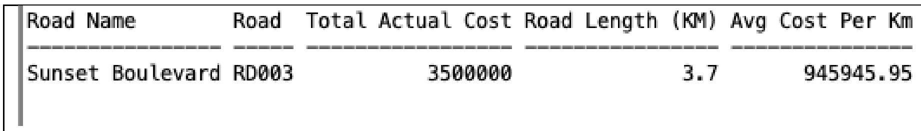
```
Employee Full Name                 Year Emp.  Assigned Project Count
---------------------------------- ---- ----- ----------------------
Alexander Johnson                  2017 E0002                      1
Benjamin Davis                     2020 E0004                      1
Emily Mitchell                     2019 E0001                      2
Olivia Martinez                    2019 E0005                      1
Sophia Williams                    2021 E0003                      1
```

Four) Purpose of query: Based on user input, retrieve specific information on a road. Prompting the user to enter a road name and then search for the details for that road, including its name, ID, total actual cost associated with projects on that road, its length in kilometers, and the average cost per kilometer.

Query:

```
 ACCEPT Road_Name CHAR PROMPT 'Enter Road Name: '

SELECT
   R.Road_Name AS "Road Name",
   R.Road_ID AS "Road ID",
   SUM(C.Contract_ActCost) AS "Total Actual Cost",
   R.Road_Kilometers AS "Road Length (KM)",
   TO_NUMBER(ROUND(AVG(C.Contract_ActCost) / R.Road_Kilometers, 2), '9999999.99') AS "Avg Cost Per Km"
FROM Road R
JOIN Project_Records PR ON R.Road_ID = PR.Road_ID
JOIN Projects P ON PR.Project_Code = P.Project_Code
JOIN Contracts C ON P.Project_Code = C.Project_Code
WHERE R.Road_Name = '&Road_Name'
GROUP BY R.Road_Name, R.Road_ID, R.Road_Kilometers;
```

Output/Result for input = Sunset Boulevard



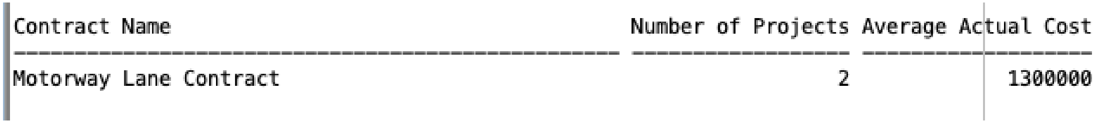
```
Road Name        Road  Total Actual Cost Road Length (KM) Avg Cost Per Km
---------------- ----- ----------------- ---------------- ---------------
Sunset Boulevard RD003           3500000              3.7       945945.95
```

Five) Purpose of query: To search for contracts with more than one project and retrieve relevant information. It retrieves the contract name, number of projects associated with that project and the average actual costs of the projects within the contract. Using left join to include contracts with no associated projects, then filtering the results to show contracts that have more than one project.

QUERY:

```
SELECT c.Contract_Name AS "Contract Name",
       COUNT(p.Project_Code) AS "Number of Projects",
       AVG(c.Contract_ActCost) AS "Average Actual Cost"
FROM Contracts c
LEFT JOIN Projects p ON c.Project_Code = p.Project_Code
GROUP BY c.Contract_Name
HAVING COUNT(p.Project_Code) > 1;
```

Output/Result:



```
Contract Name                                      Number of Projects Average Actual Cost
-------------------------------------------------- ------------------ -------------------
Motorway Lane Contract                                              2             1300000
```

An example of a query from an Employee database is given below for your reference.[2]

- **Purpose of the query:**

To search for staff with a specific qualification (e.g. BSc) and retrieve the following details (Staff Number, Staff Name (Concatenation of Last and First names) and their Qualification.

- **SQL SELECT query**

```
SELECT
        staff.staff_no, staff_fname|| ‘ ‘ || staff_lname “Staff Name”, qual_type Qualification
FROM
        staff, qualification
WHERE
        qual_type = '&Qualification'
AND
        staff.staff_no = qualification.staff_no;
```

- **Output/Result of the query**

Input for Qualification type variable = “BSC”

Staff No	Staff Name	Qualification
101	John Smith	BSc
212	Mike Bird	BSc
112	Susan Carr	BSc
117	David Cameron	BSc

[2] Please note that this is an example from a different database that is not related to your assignment.

For Part C, you must include the following (as shown above) for each SQL query statement:

- Description of the purpose of the query.
- SQL SELECT query
- The result of the query

3. Submission Instructions

You must submit a soft copy of your assignment on Canvas before the specified due date and time.

All Parts: Due Wednesday, 25th October 2023, 5:00 pm

1. **Submit the following files**:
   - DCT's Group Assignment Cover Sheet (available for download from Canvas. Please also ensure you correctly write your lab tutor's name and tutorial day and time.
   - **A PDF** or **Word file** containing your work for all the Tasks in Parts A, B, and C and your student ID and name written on the document's first page. A JPG/PNG image should be used when capturing your ERD diagram. Use appropriate formatting options (e.g., transparent background, large fonts) to ensure the diagram is clear and readable.
   - **A Visual Paradigm file** containing the ERD for Part A.

   Please ensure that each of these files (the PDF or Word, and VP) is named clearly with the surnames of both members in your group: surname1_surname2.

Please note:

- **Only one submission per group: only one student in the pair should submit.**
- You must also complete a peer evaluation survey as per details specified on Canvas.

4. Marking Scheme

Task	Marking Criteria	Max Mark
Part A Entity Relationship Modelling and Logical Database Design		
Part A – Task 1	**Identify Entities & Develop Business Rules** • Cover all the essential requirements from the case study • Uses correct syntax for business rules	10
Part A – Task 2	**Construct a logical Entity Relationship Diagram** • Identify all attributes in each entity, including all primary and foreign keys • Mapping rules applied (any composite and multivalued attributes must be transformed). • Name all relationships using verb phrases • Identify Cardinality and Participation for each relationship. • Resolve all many-to-many (M:N) relationships. • Includes any assumptions, if any.	25
Part A Total		**35**
Part B Database Implementation		
Part B – Task 3	**Oracle Table Creation Scripts for all tables in the database** • correct identification and definition of all attributes and relationships • includes the correct definition of all primary and foreign key constraints from the ERD • uses consistent naming conventions for all table and attribute names, constraints (constraints are named), and appropriate data types. • List of tables in your account	25
Part B – Task 4	**Oracle Insert Scripts for populating the tables in your database:** • Insert statements along with sample realistic test Data • the results of SELECT statements: you must run "SELECT * FROM *table_name*" (where *table_name* is the table's name) for each table in Task 3 and include the results of SELECT statements for each table.	10
Part B Total		**35**

Part C Construct SQL Queries		
Part C - Task 5	**Construct five SQL Queries:** Each query • Retrieves meaningful information to support the data and transaction requirements of the case study • each query is distinct and significantly different from the others • includes a description of the purpose of the query • Demonstrates knowledge and application of SQL content covered in the course (covers important SQL features, clauses, and functions) a. Use of restriction (WHERE clause), Arithmetic expressions, Concatenation of columns b. Use of comparison and logical operators c. Multiple table joins (e.g. Equijoin, Outer-join, and Self-join) d. Group functions (e.g., COUNT, SUM, AVG, MAX, MIN) and clauses (e.g., ORDER BY, GROUP BY, HAVING) • includes results (output rows) **Note:** • You may combine some of the preceding features in a single query. The design of each query should demonstrate your understanding of the correct usage of the various features and clauses of SQL. • A simple '**select * from table_name**' will fetch a zero mark. • There should be **no more than one query that involves a single table**	30
Part C Total		**30**
Total of Parts A, B, and C		**100**