

Week 8 Submission Tasks

Submission Process

Download **DAD_task_submission_template.docx** from the LMS.

Paste the required screen captures from the tasks below into this file.

Submit the (.docx) file into the appropriate weekly task on **the LMS**

Task 1.

Qwerty Solutions hire consultants to work on various projects. Each project has a project ID, a title and a description. Consultants all have a name, an address and a mobile phone number.

When a consultant is allocated to a project the date is recorded as well as a fixed fee. E.g. Jim was allocated to project # 63 on August 15. The fee that Jim has set for project # 63 is \$500.

Create an ERD solution based on the above narrative in draw.io or an equivalent diagramming software package.

Convert the ERD to a relational schema (DO NOT use surrogate keys)

Print or take a screenshot of the ERD and place it in the appropriate place in **DAD_task_submission_template.docx**

Copy and paste the relational schema and place it in the appropriate place in **DAD_task_submission_template.docx**

Task 2.

Welcome Services provide a number of services for clients. Examples of services are washing a car, mowing a lawn, vacuuming a house. Each service has a per hour fee. E.g. Washing a car is charged at \$35 per hour. Each service has a Description and Service Number. Each client has a name and address. Welcome Services record the date that a service was arranged for a client and the number of hours that are budget for that service. E.g. Donna Jones has arranged for lawn mowing on June 5. It is estimated that it will take 1.5 hours to complete. Note: There is no need to record who performed the service or how long it actually took to perform that task.

Create an ERD solution based on the above narrative in a suitable software package.

Convert the ERD to a relational schema (DO NOT use surrogate keys)

Print or take a screenshot of the ERD and place it in the appropriate place.

Copy and paste the relational schema and place it in the appropriate place

Task 3.

Uptown Gallery are about to host an exhibition next month. A number of works of art (paintings sculptures etc) will be exhibited. Every art work will have a barcode, a title and a description.

Each Art Work is created by a single artist. Every artist has a name, an address and a contact number.

One artist may create more than one art work for the exhibition.

People will come and visit the exhibition. As each person enters the exhibition, their name, address and

phone number will be recorded. Each person will be given a unique barcode id. During their visit, people

will nominate which art work they consider the best and this will be recorded by Uptown Gallery.

People

will be able to vote for more than one art work.

Create an ERD solution based on the above narrative in a suitable software package.

Convert the ERD to a relational schema (DO NOT use surrogate keys)

Print or take a screenshot of the ERD and place it in the appropriate place.

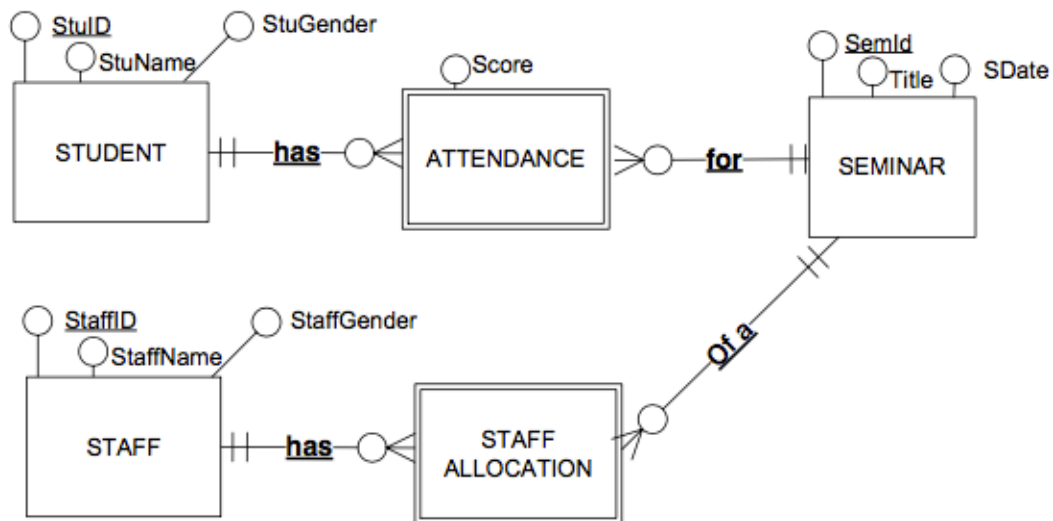
Copy and paste the relational schema and place it in the appropriate place

Task 4. (A big one)

Jellyfish Inc. are providers of training and seminar services for the IT industry.
Strangely, they currently record all of their seminar and attendance data in spreadsheets.

Very unsophisticated.

A modelling / database expert has created this ERD:



The same expert has created the following Database SQL statements to create the 'strong entity' tables as well as some sample data (retrieved from an Excel file) for testing purposes.

```
IF OBJECT_ID('STUDENT') IS NOT NULL
DROP TABLE STUDENT;
```

```
IF OBJECT_ID('STAFF') IS NOT NULL
DROP TABLE STAFF;
```

```
IF OBJECT_ID('SEMINAR') IS NOT NULL
DROP TABLE SEMINAR;
```

```
GO
```

```
CREATE TABLE STUDENT(
    STUID          INT
,   FULLNAME      NVARCHAR(50)
,   GENDER        NVARCHAR(1)
,   PRIMARY KEY (STUID)
);
```

```

CREATE TABLE STAFF(
    STFID          INT
,   FULLNAME      NVARCHAR(50)
,   GENDER        NVARCHAR(1)
,   PRIMARY KEY  (STFID)
);

CREATE TABLE SEMINAR(
    SEMID          INT
,   TITLE         NVARCHAR(50)
,   SDATE         DATETIME
,   PRIMARY KEY  (SEMID)
);

GO

INSERT INTO STUDENT (STUID, FULLNAME, GENDER) VALUES
(2719000, 'Emma Jay', 'F'),
(9091431, 'Dave Smith', 'M'),
(9198122, 'Jane Jones', 'F'),
(8184399, 'Mike Toll', 'M');

INSERT INTO STAFF (STFID, FULLNAME, GENDER) VALUES
(131, 'Karen Lovell', 'F'),
(232, 'Denise Randle', 'F'),
(455, 'Sean Mellerick', 'F'),
(217, 'Anna Langelly', 'F');

INSERT INTO SEMINAR (SEMID, TITLE, SDATE) VALUES
(401, 'Database Security', '13-JAN-2020'),
(402, 'Agile Programming', '14-JAN-2020'),
(406, 'Business Intelligence', '13-JAN-2020'),
(409, 'Social Media Analytics', '14-JAN-2020');

```

- a) You must write the DROP and CREATE TABLE statements for the Staff Allocation and Attendance tables. Ensure that you include all appropriate Primary Keys and Foreign Keys.
- b) Create SQL INSERT statements for the Staff Allocation Table based on this data retrieved from a Jellyfish Excel file. Note: You will need to determine which pieces of data from this spreadsheet are required to be inserted into the Staff Allocation table.

	A	B
1	ALLOCATION DATA	
2	Seminar Number	Staff Details
3	401	131 Karen
4	401	455 Sean
5	402	131 Karen
6	402	217 Anna
7	406	455 Sean

c) Create SQL INSERT statements for the Attendance Table based on this data retrieved from a Jellyfish Excel file. Each attendee awards a point score in the range 0-10 for the seminar that they attended. **Note:** You will need to determine which pieces of data from this spreadsheet are required to be inserted into the Attendance table (i.e. not all of it may be required in the Attendance table)

	A	B	C
1	Attendance Data		
2	Seminar Id	Student identification	The score awarded by student
3	401	9091431 Dave	8 points
4	401	9198112 Jane	5 points
5	401	8184399 Mike	7 points
6	402	9198112 Jane	6 points
7	402	9091431 Dave	7 points
8	406	9198112 Jane	4 points

d) Create SQL SELECT statements to meet the following requirements

1. Using a **UNION** clause, write an SQL statement that list the names and genders of all staff and students in a single result set. The list must be in gender / name sequence.
2. Using a **SUBQUERY** statement, write an SQL statement that displays all rows from the Attendance table only if the score given is less than the average of all scores.
3. Using an **LEFT OUTER JOIN** statement, write an SQL statement that displays a list of ALL staff names and seminar numbers of any seminars that they have presented. The list must include staff who are not allocated to any presentations.
4. Using an **RIGHT OUTER JOIN** statement, write an SQL statement that displays a list of ALL student names and shows the title of any seminar that they attended and the score they gave it. Note: This list must include students who attended no seminars.

Copy and paste the

- **Drop Table** statements (include the drop statements for Staff, Student & Seminar as well)
- **Create Table** statements (include the Create statements for Staff, Student & Seminar as well)
- **Insert** statements (include the Insert statements for Staff, Student & Seminar as well)
- **Union** statement & Result Set
- **Subquery** statement & Result Set

- **Left Outer Join** statement & Result Set
- **Right Outer Join** statement & Result Set

Place all copies **IN ORDER** int the **DAD_task_submission_template.docx**